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

Benoît Libert, Somindu Ramanna, Moti Yung

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

Benoît Libert[1], Somindu C. Ramanna[1], and Moti Yung[2]

[1] ENS de Lyon, LIP Laboratory, France
Email: benoit.libert@ens-lyon.fr, somindu.ramanna@ens-lyon.fr
[2] Snapchat and Columbia University, USA
Email: 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 $F(M)$ of the committed message during the opening phase. We provide constructions for the functionality of linear functions, where messages consist of vectors of n elements over some domain $\mathcal{D}$ (e.g., $\vec{m} = (m_1, \ldots, m_n) \in \mathcal{D}^n$) and commitments can later be opened to a specific linear function $\sum_{i=1}^{n} m_i x_i = y \in \mathcal{R}$ of the vector coordinates. An opening for a function $F : \mathcal{D}^n \to \mathcal{R}$ thus generates a witness for the fact that $F(\vec{m})$ indeed evaluates to y. One security requirement is called *function binding* and requires that it be infeasible open a commitment to two different evaluations y, y' for the same function F.

We propose a construction of functional commitment (FC) 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 FCs 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.

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

1 Introduction

Commitment schemes [8] 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 of commitment schemes are called hiding and binding. The former requires that no information about the committed 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 [35, 18], 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 assumption.

1.1 Related Works and the Open Problems

Functional commitments. Functional commitments can be seen as the natural commitment analogue of functional encryption [46, 13]. The latter primitive allows restricting what the receiver learns about encrypted data: when a decryption operation is conducted 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 committed message.

In their most general form, functional commitments were implicitly suggested by Gorbunov, Vaikuntanathan and Wichs [28] 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 inputs 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 [9]. 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 construction 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 committed message is) supporting constant-size 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. Verifiable random functions [39], 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 [26, 4] 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 [38], 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 [42] envisioned committed databases for which the sender can demonstrate more general statements than just membership of non-membership.

Vector commitments. Concise vector commitments were first suggested by Libert and Yung [35] and further developed by Catalano and Fiore [18]. They basically consist of Pedersen-like [45] 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 [19, 35] via mercurial commitments [22] supporting short coordinate-wise openings [35]. Other applications in the context of verifiable databases [7] were suggested in [18]. While concise vector commitments can be based on long-lived hardness assumptions like RSA or Computational Diffie-Hellman [18], they either require groups of hidden order (making them incompatible with zero-knowledge proofs in the standard model [29]) 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 [31], 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 [21, 27], anonymous credentials with attributes [16] or in optimized flavours of zero-knowledge databases which do not seek to hide the size of the committed set. They also imply vector commitments, as observed in [16]. Camenisch *et al.* [16] used vector commitments in a modular design of anonymous credentials where users's credentials are associated with descriptive attributes. While the commitments in [31, 16] 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 [6] 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 [6, 3, 36, 11] and includes proposals based on the Strong RSA assumption [3, 34]. The second family [41, 14], which was first explored by Nguyen [41], appeals to bilinear maps (a.k.a. pairings) and assumptions, like the Strong Diffie-Hellman assumption [10], 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 [34] (where non-membership witnesses can show that a given input was not accumulated) or dynamic accumulators [17] (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 [41, 14] 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 [2, 20], like e-cash systems, 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.* [43] 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 [43] to provide evidence that two accumulated sets have a given intersection.

A third family of accumulators [44, 11] builds on Merkle trees [37] rather than number theoretic assumptions. Its main disadvantage is that the use of hash trees entails witnesses of 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.* [25] 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 [18]. While implying solutions based on RSA or Diffie-Hellman, the generic construction of [18] 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 ([43]) 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 first generalize the notion of vector commitments (VCs) to what we call *functional commitments* (FCs) for linear functions. Similar to VCs, such a commitment scheme allows committing to vectors of messages which can later be opened to specific function evaluations. While possible [28], 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 $F_{\vec{x}}(\vec{m}) = \sum_{i=1}^{n} x_i m_i = y \in \mathcal{D}$, 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.* [30] 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 [23] 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. We now present the top level idea 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 elements $\{g^{\alpha^j}\}_{j=1}^{n}$, $\{U_j = u^{\alpha^j}\}_{j \in [1,2n] \setminus \{n+1\}}$ for some $g, u \in \mathbb{G}_{p_1}$. The trapdoor key is $U_{n+1} = u^{\alpha^{n+1}}$. Commitment to a vector $\vec{m}$ is defined as $C = g^{\gamma} \cdot \prod_{j=1}^{n} g^{\alpha^j m_j}$. Witness for a linear 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 the trapdoor $u^{\alpha^{n+1}}$ in the witness enables us to verify that $y = \langle \vec{x}, \vec{m} \rangle$ by checking whether $e(C, \prod_{i=1}^{n} u^{\alpha^{n-i+1} x_i}) = e(g^{\alpha}, u^{\alpha^n})^y \cdot e(g, W_y)$ holds. The u-components are additionally 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 simple composite-order analogue of the vector commitment scheme proposed in [35].

Proof Idea. We fix some notation first (similar to [47]). 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 [47] of the Déjà Q framework, we prove security of our FC scheme based on $(p_1 \to p_1p_2)$ and $(p_1p_3 \to p_1p_2p_3)$ 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 associated verification equations, one can say that the adversary can essentially produce $\mathit{\Delta} W = \left(W_{y'}/W_y\right)^{1/(y-y')}$ 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 $\mathit{\Delta} W$ is identical to that of the trapdoor key. Define two types of keys (attacks) according to $\{U_j\}_{j=1}^{2n}$ $(\mathit{\Delta} W)$ 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 $(p_1 \to p_1p_2)$ 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 step of the transition uses the $p_1p_3 \to p_1p_2p_3$ subgroup decision assumptions and the second transition is based on the Chinese remainder theorem (CRT) that states that $\alpha \bmod p_1$ and $\alpha \bmod p_2$ are uncorrelated. We can thus replace $\alpha \bmod p_2$ by $\alpha_1 \bmod p_2$ as long as $\alpha \bmod p_2$ is not revealed in any information provided to the attacker. By repeated application of 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 [23, 47] and hence can be replaced by a random exponent, $RF(i)$ for U_i in particular. After the final transition, creating $\mathit{\Delta} W$ 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 [18], 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.As explained above, 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 values that can be accumulated 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 $n + 1$ (the $n + 1$-st exponent being the trapdoor key). We extend this interval to $[1,(d + 1)n]$ making $n + 1, 2n + 1, \ldots, dn + 1$ part of the trapdoor. The witness component for a specific position i of the linear function was defined as $W_i = u^{\alpha^{n-i+1}\gamma} \cdot \prod_{j=1,j\neq i}^{n} u^{\alpha^{n+1+j-i} m_j}$. In order 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 a factor ℓ in the exponent. 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 $X = \{x_1, \ldots, x_k\} \not\subseteq S$ 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 $W_X/(W_{X\cap S}W_{X\setminus S})$ 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 $N = p_1p_2p_3$ 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 $g = 1_{\mathbb{G}}$. An important property of composite order groups is that 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 $\mathbb{G}_{p_ip_j}$ the subgroup of order p_ip_j. We rely on the following assumptions introduced in [33], which are non-interactive, falsifiable [40]. 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 $T \in \mathbb{G}_{p_1p_2}$ 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_1X_2, Z_3, Y_2Y_3) and T, it is hard to decide if $T \in_R \mathbb{G}_{p_1p_3}$ or $T \in_R \mathbb{G}$.

2.2 Vector Commitment Schemes

In prime order groups, Libert and Yung [35] 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 [35, 18]. In [35], 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 $r \xleftarrow{R} \mathbb{Z}_p$ 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 [12].

2.3 Functional Commitments for Linear Functions: Definitions

In [30], Izabachène *et al.* implicitly showed that the vector commitment scheme of [35] 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 $\vec{m} \cdot \vec{x} = y$, 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 $\vec{x} = (x_1, \ldots, x_n), \vec{m} = (m_1, \ldots, m_n)$. A functional commitment scheme* FC *for $(\mathcal{D}, n, \langle \cdot, \cdot \rangle)$ is a tuple of four (possibly probabilistic) polynomial time algorithms –* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$.

$\mathsf{Setup}(1^\lambda, 1^n)$**:** *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.

In some applications (e.g., [32]), it may be useful to extend the syntax with an equivocation algorithm which allows generating witnesses for arbitrary values y using the trapdoor TK. This equivocation algorithm $\mathsf{Equivocate}$ takes as input a pair (C, aux) produced as $(C, \mathsf{aux}) \leftarrow \mathsf{Commit}(CK, \vec{m})$, a vector $\vec{x} \in \mathcal{D}^n$, an arbitrary value y and the trapdoor TK. It outputs a witness W_y such that $\mathsf{Verify}(CK, C, W_y, \vec{x}, y) = 1$. While our construction readily extends to support such a mechanism, we omit it from the syntax for simplicity.

The security requirements of functional commitments are formalized as follows.

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* (Setup, Eval, WitCreate, 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.

$\mathsf{WitCreate}(PK, S, V, \mathsf{aux}, x, \mathsf{type})$**:** *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 $\mathsf{type} = 1$**:** *If $x \notin S$, it returns $\perp$. Otherwise, a membership witness W is returned.*

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

$\mathsf{Verify}(PK, V, W, x, \mathsf{type})$**:** *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 competed 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 an non-accumulated value or a non-membership witness for an accumulated value.

2.5 Accumulators Supporting Subset Queries

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* $(\mathsf{Setup}, \mathsf{Eval}, \mathsf{WitCreate}, \mathsf{Verify})$ *of probabilistic polynomial time algorithms defined as follows.*

$\mathsf{Setup}(1^\lambda, 1^n, 1^d)$**:** *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).*

$\mathsf{Eval}(PK, S)$**:** *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.

$\mathsf{WitCreate}(PK, S, V, \mathsf{aux}, X)$**:** *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.*

$\mathsf{Verify}(PK, V, W_X, X)$**:** *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

We prove that the Déjà Q framework [23] allows proving security of the functional commitment of [30] under constant size assumptions by switching to composite order groups.

$\mathbf{Setup}(1^\lambda, 1^n)$**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1 p_2 p_3$, where $p_i > 2^{l(\lambda)}$ for each $i \in \{1, 2, 3\}$, for a suitable polynomial $l : \mathbb{N} \to \mathbb{N}$. Choose $g, u \stackrel{R}{\leftarrow} \mathbb{G}_{p_1}$, $R_3 \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$ and $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_N$ at random in order to define $G_j = g^{\alpha^j}$ for each $j \in [1, n]$ and

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

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 is $TK := U_{n+1} = u^{(\alpha^{n+1})} \cdot R_{3,n+1}$, where $R_{3,n+1} \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$.

$\mathbf{Commit}(CK, \vec{m})$**:** 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 randomly chosen $\gamma \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and output the commitment C with the auxiliary information $\mathsf{aux} = (m_1, \dots, m_n, \gamma)$.

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

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

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 $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_N^n$, 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 $i \in \{1, \ldots, n\}$, (1) implies that

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

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 the commitment is perfectly hiding: since C lives in the cyclic subgroup $\mathbb{G}_{p_1}$, any vector $(m_1, \ldots, m_n) \in \mathbb{Z}_N^n$ has a corresponding opening $\gamma \in \mathbb{Z}_N$ (and even $p_2 p_3$ 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 [47, Section 4], we found it interesting to give a direct proof based on 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 1, we design an accumulator supporting subset queries in section 5. Since the latter scheme has a public key containing more elements than in [47], it can hardly be proved secure via a reduction from the security of Wee's broadcast encryption [47].

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 [47, 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 [35, 30], 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 $\{U_i\}_{i\in[1,2n]\backslash\{i\}}$, which appear in the public key. The detailed proof of Theorem 1 is given in Appendix C.

Theorem 1. *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 scheme. Indeed, in order to commit to a polynomial $P[Z] = a_0 + a_1 Z + \cdots + a_{n-1}Z^{n-1}$ of degree $n-1$, we can simply commit to the vector containing the 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 [31, 16] 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

In [18], Catalano and Fiore showed how to construct cryptographic accumulators from vector commitments. While their construction notably yields an accumulator based on the Computational Diffie-Hellman assumption, it only supports small universes. Namely, accumulated values should be taken 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 [41]), 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 simply commit to the n-dimensional vector $(a_0, a_1, \ldots, a_{n-2}, 1)$ that contains the coefficients of the polynomial $P[Z] = \prod_{j=1}^{n-1}(Z - x_j) = \sum_{j=0}^{n-2} a_j Z^j + Z^{n-1}$ 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 Appendix B.

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.* [43] 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 $N = p_1 p_2 p_3$, where $p_i > 2^{l(\lambda)}$ for each $i \in \{1, 2, 3\}$, for a suitable polynomial $l : \mathbb{N} \to \mathbb{N}$. Choose $g, u \overset{R}{\leftarrow} \mathbb{G}_{p_1}$, $R_3 \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ and $\alpha \overset{R}{\leftarrow} \mathbb{Z}_N$ 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

$$\begin{array}{lcl} U_1 = u^{\alpha} \cdot R_{3,1},\; U_2 = u^{(\alpha^2)} \cdot R_{3,2}, & \ldots & , U_n = u^{(\alpha^n)} \cdot R_{3,n} \\ U_{n+2} = u^{(\alpha^{n+2})} \cdot R_{3,n+2}, & \ldots & , U_{2n} = u^{(\alpha^{2n})} \cdot R_{3,2n}, \\ \ldots & & \\ U_{dn+2} = u^{(\alpha^{dn+2})} \cdot R_{3,dn+2}, & \ldots & , U_{(d+1)n} = u^{(\alpha^{(d+1)n})} \cdot R_{3,(d+1)n}, \end{array}$$

where $R_{3,j} \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ for each $j \in [1, (d+1)n]$. 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 $n' \leq n - 1$, 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 $k \leq d$ (we assume w.l.o.g. that $x_1, \ldots, x_k$ are arranged in some fixed lexicographical order), and the state information $\mathsf{aux} = (S, \gamma)$ such that (V, aux) was produced by $\mathsf{Eval}(PK, S)$, 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 $\vec{x}_\ell = (x_{\ell,1}, \ldots, x_{\ell,n}) = (1, x_\ell, x_\ell^2, \ldots, x_\ell^{n-1}) \in \mathbb{Z}_N^n$ which satisfies $P_S(x_\ell) = \langle \vec{m}, \vec{x}_\ell \rangle = 0$. For $\ell \in [1, k]$, generate a witness that $\langle \vec{m}, \vec{x}_\ell \rangle = 0$ by first using $\{U_{\ell n+1+j-i}\}_{j \neq i}$ to compute

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

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 $\langle \vec{m}, \vec{x}_\ell \rangle = 0$. 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 we set $d \approx n$ so as to handle queries for arbitrary subsets of size $\leq n$. In comparison with [43], 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, constant-size assumptions.

We prove that the scheme provides collision-freeness (as defined in Appendix A) in the sense that no PPT adversary can output a set S (of size $\leq n-1$) 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 the definition of collision-freeness used in [25]: 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 properly verifies.

The proof crucially 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 $X \setminus S$ by taking advantage of 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, when we adapt the proof of Theorem 1, this forces the adversary to predict a linear combination of random function evaluations (which is statistically unpredictible) in the final step of the sequence of games.

Theorem 2. *The scheme is collision-free if Assumption 1 and Assumption 2 hold.* (The proof is available in Appendix D.)

Acknowledgements

The first two authors were 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.

References

1. T. Acar, L. Nguyen. Revocation for Delegatable Anonymous Credentials. In *PKC'11*, *LNCS* 6571, pp. 423–440, 2011.
2. M. H. Au, Q. Wu, W. Susilo, and Y. Mu. Compact E-Cash from Bounded Accumulator. In *CT-RSA 2007*, volume 4377 of *LNCS*, pages 178–195. Springer, 2007.
3. N. Baric and B. Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494. Springer, 1997.
4. M. Bellare, D. Hofheinz, S. Yilek. Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening. In *Eurocrypt*'09, *LNCS* 5479, pp. 1–35, 2009.
5. M. Bellare, P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM Press, 1993.
6. J. Benaloh, M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In *Eurocrypt'93*, *LNCS* 4948, pp. 274–285, 1993.
7. S. Benabbas, R. Gennaro, Y. Vahlis. Verifiable Delegation of Computation over Large Datasets. In *Crypto 2011*, pp. 111–131, *LNCS* 6841, Springer, 2011.
8. M. Blum. Coin Flipping by Telephone. In *Crypto*'81, pp. 11-15, 1981.
9. M. Blum, J. Feldman, S. Micali. Non-Interactive Zero-Knowledge and its Applications. In *STOC*'88, pp. 103–112, 1988.

10. D. Boneh, X. Boyen. Short Signatures Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 56–73. Springer-Verlag, 2004.
11. D. Boneh and H. Corrigan-Gibbs. Bivariate Polynomials Modulo Composites and Their Applications. In *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 42–62. Springer, 2014.
12. D. Boneh, C. Gentry and B. Waters. Collusion-Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *Crypto'05*, *LNCS* 3621, pp. 258–275, 2005.
13. D. Boneh, A. Sahai, B. Waters. Functional Encryption: Definitions and Challenges. In *TCC'11*, *LNCS* 6597, pp. 253–273, 2011.
14. J. Camenisch, M. Kohlweiss, C. Soriente. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC'09*, *LNCS* 5443, pp. 481–500, 2009.
15. J. Camenisch, M. Kohlweiss, C. Soriente. Solving Revocation with Efficient Update of Anonymous Credentials. In *SCN'10*, *LNCS* 6280, pp. 454–471, 2010.
16. J. Camenisch, M. Dubovitskaya, K. Haralambiev, M. Kohlweiss. Composable & Modular Anonymous Credentials: Definitions and Practical Constructions. In *Asiacrypt'15*, *LNCS* 9453, pp. 262–288, 2015.
17. J. Camenisch, A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto'02*, *LNCS* 2442, pp. 61–76, Springer, 2002.
18. D. Catalano, D. Fiore. Vector Commitments and their Applications. In *PKC* 2013, *LNCS* 7778, pp. 55–72, 2013.
19. D. Catalano, D. Fiore, M. Messina. Zero-Knowledge Sets with Short Proofs. In *Eurocrypt*'08, *LNCS* 4965, pp. 433–450, 2008.
20. S. Canard and A. Gouget. Multiple Denominations in E-cash with Compact Transaction Data. In *FC 2010*, volume 6052 of *LNCS*, pages 82–97. Springer, 2010.
21. B. Chor, S. Goldwasser, S. Micali, B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In *FOCS*'85, pp. 383–395, 1985.
22. M. Chase, A. Healy, A. Lysyanskaya, T. Malkin, L. Reyzin. Mercurial Commitments with Applications to Zero-Knowledge Sets. In *Eurocrypt'05*, *LNCS* 3494, pp. 422–439, Springer, 2005.
23. M. Chase, S. Meiklejohn. Déjà Q: Using Dual Systems to Revisit q-Type Assumptions In *Eurocrypt 2014*, *LNCS* 8441, pp. 622–639, Springer, 2014.
24. I. Damgård, N. Triandopoulos. Supporting Non-membership Proofs with Bilinear-map Accumulators. In Cryptology ePrint Archive: Report 2008/538.
25. D. Derler, C. Hanser, D. Slamanig. Revisiting Cryptographic Accumulators, Additional Properties and Relations to Other Primitives. In *CT-RSA 2015*, *LNCS* 9048, pp. 127-144, Springer, 2015.
26. C. Dwork, M. Naor, O. Reingold, L. Stockmeyer. Magic Functions. In *FOCS*'99, pp. 523–534, 1999.
27. P. Feldman. A Practical Scheme for Non-interactive Verifiable Secret Sharing. In *FOCS*'87, pp. 427–437, 1987.
28. S. Gorbunov, V. Vaikuntanathan, D. Wichs. Leveled Fully Homomorphic Signatures from Standard Lattices. In *STOC 2015*, pp. 469–477, 2015.
29. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
30. M. Izabachène, B. Libert, D. Vergnaud. Blockwise P-Signatures and Non-Interactive Anonymous Credentials with Efficient Attributes. In *IMACC 2011*, pp. 431–450, Springer, 2011.
31. A. Kate, G. Zaverucha, I. Goldberg. Constant-Size Commitments to Polynomials and their Applications. In *Asiacrypt 2010*, pp. 177–194, LNCS 6477, Springer, 2010.
32. J. Krupp, D. Schröder, M. Simkin, D. Fiore, G. Ateniese, S. Nuernberger. Nearly Optimal Verifiable Data Streaming. In *PKC 2016*, LNCS series, pp. 417–445, LNCS 9614, 2016.
33. A. Lewko, B. Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010*, LNCS 5978, Springer, 2010.

34. J. Li, N. Li, and R. Xue. Universal Accumulators with Efficient Nonmembership Proofs. In *ACNS 2007*, volume 4521 of *LNCS*, pages 253–269. Springer, 2007.
35. B. Libert and M. Yung. Concise Mercurial Vector Commitments and Independent Zero-Knowledge Sets with Short Proofs. In *TCC 2010*, LNCS 5978, pp. 499–517, 2010.
36. H. Lipmaa. Secure Accumulators from Euclidean Rings Without Trusted Setup. In *ACNS 2012*, volume 7341 of *LNCS*, pages 224–240. Springer, 2012.
37. R. C. Merkle. A Certified Digital Signature. In *CRYPTO 1989*, volume 435 of *LNCS*, pages 218–238. Springer, 1989.
38. S. Micali, M.-O. Rabin, J. Kilian. Zero-Knowledge Sets. In *FOCS 2003*, pp. 80–91, 2003.
39. S. Micali, M.-O. Rabin, S. Vadhan. Verifiable Random Functions. In *FOCS 1999*, pp. 120–130, 1999.
40. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109. Springer-Verlag, 2003.
41. L. Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA'05*, *LNCS* 3376, pp. 275–292, 2005.
42. R. Ostrovsky, C. Rackoff, A. Smith. Efficient Consistency Proofs for Generalized Queries on a Committed Database. In *ICALP'04*, *LNCS* 3142, pp. 1041–1053, 2004.
43. C. Papamanthou, R. Tamassia, N. Triandopoulos. Optimal Verification of Operations on Dynamic Sets. In *Crypto 2011*, *LNCS* 6841, pp. 91–110, 2011.
44. C. Papamanthou, E. Shi, R. Tamassia, and K. Yi. Streaming Authenticated Data Structures. In *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 353–370. Springer, 2013.
45. T. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Crypto'91*, *LNCS* 576, pp. 129–140, 1991.
46. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt'05*, *LNCS* 3494, pp. 457–473, 2005.
47. H. Wee. Déjà Q Encore! Un petit IBE. In *TCC 2016*, *LNCS* 9563, pp. 237–258, 2016.

A Security Definitions for Cryptographic Accumulators

This section recalls two important security notions for accumulators.

Informally undeniability requires that no PPT attacker be able to generate both membership and non-membership witnesses for any element of the domain.

Definition 6 (Undeniability). *A cryptographic accumulator is* **undeniable** *if every PPT adversary $\mathcal{A}$ has negligible advantage in winning the following game.*

- *The challenger generates a pair of keys (PK, SK) by running the* Setup *algorithms and provides PK to the adversary $\mathcal{A}$.*
- *The adversary $\mathcal{A}$ eventually halts and outputs $(V, x, W, \widehat{W})$. It wins if the conditions* $\mathsf{Verify}(PK, V, W, x, 1) = 1$ *and* $\mathsf{Verify}(PK, V, \widehat{W}, x, 0) = 1$ *are simultaneously satisfied.*

As shown by [25], the notion of undeniability implies that of collision-freeness and is actually strictly stronger. It was shown that there exist accumulators that are collision-free but are not undeniable.

The notion of indistinguishability captures that the accumulations of any two distinct sets be computationally indistinguishable.

Definition 7 (Indistinguishability). *An accumulator if for any PPT adversary $\mathcal{A}$ has negligible advantage in winning the following game.*

- *The challenger generates a pair of keys (PK, SK) by running the* Setup *algorithms and provides PK to the adversary $\mathcal{A}$.*

- *$\mathcal{A}$ returns two sets S_0, S_1. The challenger picks a random bit $\beta \in \{0,1\}$, computes $(V, \mathsf{aux}) \leftarrow \mathsf{Eval}(PK, S_b)$ and provides V to $\mathcal{A}$.*
- *$\mathcal{A}$ returns a bit β'. $\mathcal{A}$ wins the game if $\beta = \beta'$.*

Definition 8 (Collision-Freeness). *An accumulator supporting subset queries is collision free if every PPT adversary $\mathcal{A}$ has negligible advantage in winning the following game.*

- *The challenger generates a pair of keys (PK, SK) by running the* Setup *algorithms and provides PK to the adversary $\mathcal{A}$.*
- *$\mathcal{A}$ outputs $(S, \mathsf{aux}, X, W_X)$ with $|S| \leq n$, $|X| \leq d$ and wins if* $\mathsf{Verify}(PK, V, W_X, X) = 1$ *and $X \not\subset S$, where V is the accumulator value produced by* Eval *on input of X and the randomness contained in* aux.

B A Concrete Universal Accumulator based on Subgroup Decision Assumptions

Our universal accumulator uses a randomized evaluation algorithm so as to achieve the indistinguishability property of [25]. It can be made deterministic setting $\gamma = 0$. The construction is a universal accumulator in that it supports both membership and non-membership witnesses. The scheme can be seen as a variant of Nguyen's accumulator [41] and goes as follows.

Gen$(1^\lambda, 1^n)$**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1 p_2 p_3$, where $p_i > 2^{l(\lambda)}$ for each $i \in \{1, 2, 3\}$, for a suitable polynomial $l : \mathbb{N} \rightarrow \mathbb{N}$. 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 in order to define

$$G_1 = g^{\alpha}, \qquad G_2 = g^{(\alpha^2)}, \qquad \dots \qquad , G_n = g^{(\alpha^n)}$$

and

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

where $R_{3,j} \xleftarrow{R} \mathbb{G}_{p_3}$ for each $j \in [1, 2n] \backslash \{n+1\}$. Define the public key as

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

The secret key is $SK := \alpha$.

Eval(PK, S)**:** To hash a set $S = \{y_1, \dots, y_{n'}\}$ of cardinality $n' \leq n-1$, define the polynomial $P[Z] = \prod_{j=1}^{n'} (Z - y_j) = \sum_{j=0}^{n'} a_j \cdot Z^j$ (where $a_{n'} = 1$) and the vector $\vec{m} = (m_1, \dots, m_n) = (a_0, a_1, \dots, a_{n'-1}, 1, 0, \dots, 0) \in \mathbb{Z}_N^n$. Then, choose $\gamma \xleftarrow{R} \mathbb{Z}_N$ and compute

$$V = g^{\gamma} \cdot \prod_{j=1}^{n} G_j^{m_j} = g^{\gamma + \alpha \cdot P(\alpha)} \tag{4}$$

and output V and $\mathsf{aux} = (m_1, \dots, m_n, \gamma)$.

WitCreate$(PK, V, S, \mathsf{aux}, x, \mathsf{type})$**:** Given a set $S = \{y_1, \ldots, y_{n'}\}$ and the state information $\mathsf{aux} = (y_1, \ldots, y_{n'}, \gamma)$ such that (V, aux) was produced by $\mathsf{Acc}(PK, S)$ and a Boolean value $\mathsf{type} \in \{0, 1\}$, do the following:

- If $\mathsf{type} = 1$, return $\perp$ if $x \notin S$. Otherwise, a membership witness is generated as a witness showing that V is a deterministic commitment to $\vec{m} = (a_0, a_1, \ldots, a_{n'-1}, 1, 0, \ldots, 0)$ such that $\vec{m} \cdot (1, x, x^2, \ldots, x^{n-1}) = 0$. Namely, for the linear function $\vec{m} \cdot \vec{x} = \sum_{i=1}^{n} m_i \cdot x_i$ by computing

$$W_i = U_{n-i+1}^{\gamma} \prod_{j=1, j \neq i}^{n'-1} U_{n+1+j-i}^{a_{j-1}} \cdot U_{n+1+n'-i} \qquad \forall i \in \{1, \ldots, n\},$$

and outputting the witness

$$W = \prod_{i=1}^{n} W_i^{(x^{i-1})}.$$

- If $\mathsf{type} = 0$, return $\perp$ if $x \in S$. Otherwise, a non-membership witness obtained by first defining the vector $\vec{m} = (a_0, a_1, \ldots, a_{n'-1}, 1, 0, \ldots, 0)$ containing the coefficients of $P[Z] = \sum_{j=0}^{n'-1} a_j Z^j + Z^{n'} = \prod_{u \in S}(Z - u)$. The first part of the witness is the value $w_x = \vec{m} \cdot (1, x, x^2, \ldots, x^{n-1}) = P(x)$ and the witness W_x showing that $\vec{m} \cdot \vec{x} = w_x$. The non-membership witness (w_x, W_X) is returned.

Verify$(PK, V, W, x, \mathsf{type})$**:** Given an accumulator value $V \in \mathbb{G}$, a witness W, an element x of the universe $\mathbb{Z}_N$ and a bit $\mathsf{type} \in \{0, 1\}$, do the following:

- If $\mathsf{type} = 1$, parse the witness as $W \in \mathbb{G}$ (and return 0 if it does not parse properly), define the vector $\vec{x} = (1, x, x^2, \ldots, x^{n-1}) \in \mathbb{Z}_N^n$ and return 1 if and only if

$$e(V, \prod_{i=1}^{n} U_{n-i+1}^{(x^{i-1})}) = e(g, W). \tag{5}$$

Otherwise, return 0.

- If $\mathsf{type} = 0$, parse the witness W as $(w_x, W_x) \in \mathbb{Z}_N \times \mathbb{G}$ and output 0 if it does not parse properly. Using $x \in \mathbb{Z}_N$, define $\vec{x} = (1, x, x^2, \ldots, x^{n-1}) \in \mathbb{Z}_N^n$ and return 1 if and only if $y \neq 0$ and

$$e(V, \prod_{i=1}^{n} U_{n-i+1}^{(x^{i-1})}) = e(G_1, U_n)^{w_x} \cdot e(g, W_x). \tag{6}$$

Otherwise, return 0.

From relation (4), we immediately see the similarity between Nguyen's accumulator [41] and ours. In both schemes, the public key contains $\{g^{(\alpha^i)}\}_{i=0}^{n}$ and, in the deterministic version of our scheme (i.e., when $\gamma = 0$), the accumulator value is generated in the same way. The main difference is that, by introducing $O(n)$ additional elements $\{U_i\}_{i \in [1,2n] \setminus \{n+1\}}$ in the public key (which only increases its length by a small constant factor), we can generate witnesses in a different way.

Theorem 3. *The above universal accumulator is unconditionally indistinguishable and provides undeniability if Assumption 1 and Assumption 2 hold.*

Proof. The proof of indistinguishability is straightforward. As for the undeniability property, let us assume that, on input of a public key PK, an adversary $\mathcal{A}$ can produce an accumulator value $V \in \mathbb{G}$ and an element $x \in \mathbb{Z}_N$ for which it manages to output accepting membership and non-membership witnesses W and (w_x, W_x) that satisfy (5) and (6), respectively. Such an adversary immediately implies an algorithm $\mathcal{B}$ that breaks the binding property of the functional commitment in Section 3. In the game of Definition 3, the binding adversary $\mathcal{B}$ outputs the vector $\vec{x} = (1, x, x^2, \ldots, x^{n-1}) \in \mathbb{Z}_N^n$ and witnesses W, W_x, which convincingly prove $\vec{m} \cdot \vec{x} = 0$, $\vec{m} \cdot \vec{x} = x_x$, respectively. □

C Proof of Theorem 1

Proof. Recall that, in order to break the binding property of the scheme, an adversary $\mathcal{A}$ must come up with a commitment $C \in \mathbb{G}$, a vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_N^n$ and two distinct values $y, y' \in \mathbb{Z}_N$ with witnesses $W_y, W_{y'} \in \mathbb{G}$ such that

$$
\begin{aligned}
e(C, \prod_{i=1}^{n} U_{n-i+1}^{x_i}) &= e(G_1, U_n)^y \cdot e(g, W_y), \qquad (7)\\
e(C, \prod_{i=1}^{n} U_{n-i+1}^{x_i}) &= e(G_1, U_n)^{y'} \cdot e(g, W_{y'}).
\end{aligned}
$$

By dividing the two equations of (7), we find that $e(G_1, U_n)^{y-y'} = e(g, W_{y'}/W_y)$. We first assume that $y' \neq y \bmod p_i$ for each $i \in \{1, 2, 3\}$ since, otherwise, the reduction would be able to find a non-trivial factor of N, which would contradict either Assumption 1 or Assumption 2. If $\gcd(y' - y, N) = 1$, $\Delta W = \big(W_{y'}/W_y\big)^{1/(y-y')}$ is thus of the form $\Delta W = u^{(\alpha^{n+1})} \cdot g_2^{r_2} \cdot g_3^{r_3}$, for some $r_2 \in \mathbb{Z}_{p_2}$ and $r_3 \in \mathbb{Z}_{p_3}$. Note that ΔW can be seen as a "semi-functional" trapdoor in that it is equivalent to a product of the normal trapdoor TK with a $\mathbb{G}_{p_2}$ component.

In the proof, we will distinguish two kinds of attacks against the binding property.

Type A attacks: are those for which $\Delta W = \big(W_{y'}/W_y\big)^{1/(y-y')}$ lives in the subgroup $\mathbb{G}_{p_1 p_3}$.

Type B attacks: are such that ΔW has a $\mathbb{G}_{p_2}$ component (i.e., $r_2 \neq 0$) and is thus a semi-functional trapdoor.

The proof proceeds via a sequence of games involving alternative distributions of the commitment key CK and the trapdoor TK. To define these alternative distributions, it will be convenient to define a family of functions $\{F_k : [1, 2n] \to \mathbb{Z}_{p_2}\}_{k=0}^{2n}$ such that

$$
F_k(i) = \sum_{j=1}^{k} r_j \cdot \alpha_j^i \bmod p_2 \qquad \forall k \in [1, 2n], \qquad\qquad F_0(i) = 0 \qquad \forall i \in [1, 2n],
$$

where $r_1, \ldots, r_{2n}, \alpha_1, \ldots, \alpha_{2n} \overset{R}{\leftarrow} \mathbb{Z}_{p_2}$ are chosen at random by the challenger that generates CK for the adversary. Having defined the function family $\{F_k(.)\}_{k=1}^{2n}$, we can further consider several sub-classes of Type B attacks, where the semi-functional component of ΔW is determined by $\{F_k(.)\}_{k=1}^{2n}$:

Type B.k attacks ($0 \leq k \leq 2n$): are such that $\Delta W = \big(W_{y'}/W_y\big)^{1/(y-y')}$ is of the form $\Delta W = u^{(\alpha^{n+1})} \cdot g_2^{F_k(n+1)} \cdot S_3$, for some $S_3 \in \mathbb{G}_{p_3}$. In this case, we say that ΔW is a Type B.k semi-functional trapdoor.

Using the function family $\{F_k\}_{k=1}^{2n}$, we finally define gradually modified distributions for the commitment key CK and the trapdoor $TK := U_{n+1}$.

Type k parameters ($0 \leq k \leq 2n$): are parameters where elements $\{U_i\}_{i\in[1,2n]}$ now have a $\mathbb{G}_{p_2}$ component determined by the function $F_k(.)$: namely,

$$U_i = u^{(\alpha^i)} \cdot g_2^{F_k(i)} \cdot R_{3,i} \qquad \forall i \in [1, n].$$

These elements induce a modified joint distribution of CK, which contains the group elements $\{U_i\}_{i\in[1,2n]\setminus\{n+1\}}$, and $TK = U_{n+1}$.

For convenience, we define Type B.0 attacks and Type 0 keys (CK, TK) as being identical to Type A attacks and normal keys (i.e., distributed like those of the real scheme), respectively.

The sequence of games begins with the real attack game, where the adversary is given a normal commitment key (with the same distribution as in the real scheme). Then, we gradually modify the distribution of CK and prove that, unless either Assumption 1 or Assumption 2 is false, the adversary will produce an attack of Type B.k when fed with parameters of Type k. In the last game, CK consists of Type B.$2n$ parameters and we argue that the adversary's advantage in producing a Type B.$2n$ attack is statistically negligible.

For each index $0 \leq k \leq 2n$, we denote by win_k the event that the adversary wins in Game k. We also define E_k to be the event that $\mathcal{A}$ mounts a Type B.k attack when the commitment key CK is generated using Type k parameters.

Game 0: The adversary $\mathcal{A}$ receives a commitment key CK which is as in the real scheme. In Appendix C, Lemma 1 shows that, if Assumption 1 holds, any PPT adversary cannot produce anything but a Type A attack. Namely, $\Pr[\mathsf{win}_0 \wedge \neg E_0] \leq \mathbf{Adv}^1_{\mathcal{B}}(\lambda)$, where $\mathbf{Adv}^1_{\mathcal{B}}(\lambda)$ denotes $\mathcal{B}$'s advantage in breaking Assumption 1.

Since $\Pr[\mathsf{win}_0] = \Pr[\mathsf{win}_0 \wedge E_0] + \Pr[\mathsf{win}_0 \wedge \neg E_0]$, we are left with the task of bounding the term $\Pr[\mathsf{win}_0 \wedge E_0]$. To this end, we will show that, if Assumption 2 holds, $\Pr[\mathsf{win}_0 \wedge E_0]$ is negligibly different from $\Pr[\mathsf{win}_{2n} \wedge E_{2n}]$, which is negligible.

Game k ($1 \leq k \leq 2n$): The commitment key CK and the trapdoor $TK := U_{n+1}$ now have a modified distribution obtained by having the challenger generate Type k parameters before giving CK to $\mathcal{A}$. Specifically, elements $\{U_j\}_{j\in[1,2n]}$ now have a $\mathbb{G}_{p_2}$ component determined by the function $F_k(.)$:

$$U_i = u^{(\alpha^i)} \cdot g_2^{F_k(i)} \cdot R_{3,i} \qquad \forall i \in [1, n]$$

Lemma 2 shows that, under Assumption 2, the probability that $\mathcal{A}$'s attack reveals a semi-functional TK of the same type as CK is about the same in Game $k-1$ and Game k. Said otherwise, $|\Pr[\mathsf{win}_k \wedge E_k] - \Pr[\mathsf{win}_{k-1} \wedge E_{k-1}]| \leq \mathbf{Adv}^2_{\mathcal{B}}(\lambda)$.

We conclude the proof by arguing that $\Pr[\mathsf{win}_{2n} \wedge E_{2n}] \leq 1/p_2$, which is negligible. To see this, it suffices to observe that $F_{2n} : [1, 2n] \to \mathbb{Z}_{p_2}$ is a random function in the adversary's view, as shown in [47, Theorem 2]. Hence, conditionally on $\{F_{2n}(j)\}_{j\in[1,2n]\setminus\{n+1\}}$, the function evaluation $F_{2n}(n + 1)$ is uniformly distributed over $\mathbb{Z}_{p_2}$. □

Lemma 1. *In Game* 0*, any adversary producing a Type B attack implies a distinguisher for Assumption* 1*. We have* $\Pr[\mathsf{win}_0 \wedge \neg E_0] \leq \mathbf{Adv}^1_{\mathcal{B}}(\lambda)$.

Proof. Let $\mathcal{A}$ be an adversary that mounts a Type B attack when fed with a commitment key CK of Type A. We build an algorithm $\mathcal{B}$ that takes as input $(g \in \mathbb{G}_{p_1},\ X_3 \in \mathbb{G}_{p_3}, T)$ and finds an element η of $\mathbb{G}_{p_2 p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component. In turn, such an $\eta \in \mathbb{G}_{p_2 p_3}$ allows deciding whether $T \in \mathbb{G}_{p_1}$ or $T \in \mathbb{G}_{p_1 p_2}$ since $e(\eta, T) = 1_{\mathbb{G}_T}$ when $T \in \mathbb{G}_{p_1}$.

Algorithm $\mathcal{B}$ can faithfully generate the commitment key CK using its input elements $g \in \mathbb{G}_{p_1}$ and $X_3 \in \mathbb{G}_{p_3}$. By hypothesis, $\mathcal{A}$ outputs a commitment C, two distinct values $y, y' \in \mathbb{Z}_N$ and their corresponding witnesses $W_y, W_{y'} \in G$ such that relations (7) are satisfied and $\varDelta W = \big(W_{y'}/W_y\big)^{1/(y-y')}$ has a non-trivial $\mathbb{G}_{p_2}$ component. Moreover, $\mathcal{B}$ can cancel out the $\mathbb{G}_{p_1}$ component of $\varDelta W$ by computing $\eta = \varDelta W / u^{(\alpha^{n+1})}$, which is indeed an element of $\mathbb{G}_{p_2 p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component. At this point, $\mathcal{B}$ returns 1 (meaning that $T \in \mathbb{G}_{p_1}$) if $e(\eta, T) = 1_{\mathbb{G}_T}$ and 0 otherwise. □

Lemma 2. *Under Assumption* 2*, the probability of $\mathcal{A}$'s attack to be a Type B.k attack in Game k is negligibly far apart from its probability of being a Type B.*$(k-1)$ *attack in Game* $k-1$*. Concretely, there exists a distinguisher $\mathcal{B}$ running in about the same time as $\mathcal{A}$ and such that*

$$|\Pr[\mathsf{win}_k \wedge E_k] - \Pr[\mathsf{win}_{k-1} \wedge E_{k-1}]]| \leq \mathbf{Adv}^2_{\mathcal{B}}(\lambda).$$

Proof. Let us assume that there exist an index $k \in [1, 2n]$ and an adversary $\mathcal{A}$ such that $\epsilon = |\Pr[\mathsf{win}_k \wedge E_k] - \Pr[\mathsf{win}_{k-1} \wedge E_{k-1}]]|$ is non-negligible. We build a distinguisher $\mathcal{B}$ with advantage $\geq \epsilon$ against Assumption 2.

Algorithm $\mathcal{B}$ takes as input $(g, X_1X_2, Z_3, Y_2Y_3, T)$ and uses $\mathcal{A}$ to decide if $T \in_R \mathbb{G}_{p_1 p_3}$ or $T \in_R \mathbb{G}$. To this end, $\mathcal{B}$ generates the commitment key CK and the trapdoor TK as follows. It picks $\alpha \overset{R}{\leftarrow} \mathbb{Z}_N$ and defines

$$G_i = g^{(\alpha^i)} \qquad \forall i \in [1, n].$$

It also chooses $\alpha_1, \ldots, \alpha_{k-1}, r_1, \ldots, r_{k-1} \overset{R}{\leftarrow} \mathbb{Z}_N$ and computes

$$U_i = T^{(\alpha^i)} \cdot (Y_2Y_3)^{\sum_{j=1}^{k-1} r_j \cdot \alpha_j^i} \cdot R_{3,i} \qquad \forall i \in [1, 2n],$$

for randomly drawn $R_{3,i} \overset{R}{\leftarrow} \mathbb{G}_{p_3}$, so that u is implicitly defined to be the $\mathbb{G}_{p_1}$ component of T. The commitment key $CK = \big(g, \{G_j\}_{j=1}^n, \{U_j\}_{j \in [1,2n] \setminus \{n+1\}}, Z_3\big)$ is given to $\mathcal{A}$ while $\mathcal{B}$ keeps the trapdoor $TK := U_{n+1}$ to itself. Then, $\mathcal{A}$ is expected to output $C \in \mathbb{G}$, $y, y' \in \mathbb{Z}_N$ and $W_y, W_{y'} \in \mathbb{G}$ such that $y \neq y'$ and which satisfy relations (7). At this point, $\mathcal{B}$ computes $\varDelta W = \big(W_{y'}/W_y\big)^{1/(y-y')}$, which must be of the form $\varDelta W = u^{(\alpha^{n+1})} \cdot Y_2^{r_2} \cdot Y_3^{r_3}$. From $\varDelta W$, checks whether the equality

$$e(X_1X_2, \varDelta W / U_{n+1}) = 1_{\mathbb{G}_T} \tag{8}$$

holds, which means that $\varDelta W$ and U_{n+1} are identical in their $\mathbb{G}_{p_1 p_2}$ component. If so, it also means that $\varDelta W$ is a trapdoor of the same type as the commitment key CK. If (8) is indeed satisfied, $\mathcal{B}$ thus outputs 1. Otherwise, it outputs 0.

We remark that, if $T \in_R \mathbb{G}_{p_1 p_2}$, then $\mathcal{B}$ is playing Game $k-1$ with $\mathcal{A}$ since we have

$$U_i = u^{(\alpha^i)} \cdot Y_2^{\sum_{j=1}^{k-1} r_j \cdot \alpha_j^i} \cdot \tilde{R}_{3,i} \qquad \forall i \in [1, 2n],$$

for some random $\tilde{R}_{3,i} \in_R \mathbb{G}_{p_3}$. If $T \in_R \mathbb{G}$, it can be written $T = u \cdot Y_2^{s_2} \cdot Y_3^{s_3}$, so that we have

$$U_i = u^{(\alpha^i)} \cdot Y_2^{s_2 \cdot \alpha^i + \sum_{j=1}^{k-1} r_j \cdot \alpha_j^i} \cdot \tilde{R}_{3,i} \qquad \forall i \in [1, 2n],$$

with uniformly random $\tilde{R}_{3,i} \in_R \mathbb{G}_{p_3}$. In this case, $\mathcal{A}$'s view is identical to its view in Game k, where $r_k = s_2 \bmod p_2$ and $\alpha_k = \alpha \bmod p_2$ (note that $\alpha \bmod p_2$ is uncorrelated to $\alpha \bmod p_1$, so that $\alpha \bmod p_2$ does not appear anywhere but in $\{U_i\}_{i \neq n+1}$).

As a consequence, if moving from Game $k-1$ to Game k significantly increases $\mathcal{A}$'s probability of mounting an attack of a different type than CK, then $\mathcal{B}$ outputs 1 with noticeably different probabilities when $T \in_R \mathbb{G}_{p_1 p_3}$ and $T \in_R \mathbb{G}$. This clearly contradicts Assumption 2. □

D Proof of Theorem 2

Proof. To break the collision-freeness of the scheme, the adversary must produce a set $S = \{y_1, \ldots, y_{n'}\}$ of size $n' \leq n-1$, another set $X = \{x_1, \ldots, x_k\}$ such that $X \not\subseteq S$, an exponent $\gamma \in \mathbb{Z}_N$, and a witness W_X such that

$$e(V, \prod_{\ell=1}^{k} \prod_{i=1}^{n} U_{\ell n+1-i}^{(x_\ell^{i-1})}) = e(g, W_X), \tag{9}$$

where

$$\vec{x}_\ell = (x_{\ell,1}, \ldots, x_{\ell,n}) = (1, x_\ell, x_\ell^2, \ldots, x_\ell^{n-1}) \qquad \forall \ell \in [1, k]$$

and V is computed by defining the polynomial $P_S[Z] = \prod_{j=1}^{n'}(Z - y_j) = \sum_{j=0}^{n'} m_j Z^j$ and its corresponding vector $\vec{m} = (m_0, \ldots, m_{n'}, 0, \ldots, 0) \in \mathbb{Z}_N^n$ before computing the accumulator value $V = g^\gamma \cdot \prod_{j=1}^{n'+1} G_j^{m_{j-1}}$.

By hypothesis, we know that there exists $t \in [1, k]$ such that $x_t \in X \setminus S$. For each $x_t \in X \setminus S$, we assume that there exists no element $y_i \in S$ such that $x_t = y_i \bmod p_1$, but $x_t \neq y_i \bmod p_2$ or $x_t \neq y_i \bmod p_3$. Otherwise, a non-trivial factor of N would be exposed.

For each $x_t \in X \setminus S$ (where t denotes the position of x_t in X in lexicographical order), we know that the vector $\vec{m} = (m_0, \ldots, m_{n'}, 0, \ldots, 0) \in \mathbb{Z}_N^n$ satisfies $w_t = \langle \vec{m}, \vec{x}_t \rangle \neq 0 \bmod N$. We can assume w.l.o.g. that $w_t = \langle \vec{m}, \vec{x}_t \rangle \neq 0 \bmod p_2$ since, otherwise, a factor of N would be extractable.

Knowing the vector $\vec{m} = (m_0, \ldots, m_{n'}, 0, \ldots, 0) \in \mathbb{Z}_N^n$ and the adversarially-supplied randomness $\gamma \in \mathbb{Z}_N$ such that $V = g^\gamma \cdot \prod_{j=1}^{n'+1} G_j^{m_{j-1}}$, for each such $x_t \in X \setminus S$, the reduction $\mathcal{B}$ can compute $W_t \in \mathbb{G}$ such that

$$e(V, \prod_{i=1}^{n} U_{tn+1-i}^{(x_t^{i-1})}) = e(G_1, U_{tn})^{w_t} \cdot e(g, W_t), \tag{10}$$

(where $t \in [1,k]$ is the position of x_t in X in lexicographical order) so that their product $W_{X\setminus S} = \prod_{\{t:x_t\in X\setminus S\}} W_t$ satisfies

$$e\big(V, \prod_{t:x_t\in X\setminus S}\prod_{i=1}^{n} U_{tn+1-i}^{(x_t^{i-1})}\big) = \prod_{t:x_t\in X\setminus S} e(G_1, U_{tn})^{w_t} \cdot e(g, W_{X\setminus S}). \tag{11}$$

Moreover, for each ℓ such that $x_\ell \in X \cap S$ (where $\ell \in [1,k]$ is the position of x_ℓ in X in lexicographical order), the reduction can compute a witness W_ℓ such that

$$e(V, \prod_{i=1}^{n} U_{\ell n+1-i}^{x_\ell^{(i-1)}}) = e(G_1, U_{\ell n})^0 \cdot e(g, W_\ell),$$

since $\langle \vec{m}, \vec{x}_\ell \rangle = 0 \bmod N$. If we compute the witness product $W_{X\cap S} = \prod_{\{\ell:x_\ell\in X\cap S\}} W_\ell$, it thus satisfies

$$e\big(V, \prod_{\ell:x_\ell\in X\cap S}\prod_{i=1}^{n} U_{\ell n+1-i}^{(x_\ell^{i-1})}\big) = e(g, W_{X\cap S}). \tag{12}$$

Dividing (12) from (9), the reduction $\mathcal{B}$ can compute $\tilde{W}_{X\setminus S} = W_X / W_{X\cap S}$ such that

$$e\big(V, \prod_{\ell:x_\ell\in X\setminus S}\prod_{i=1}^{n} U_{\ell n+1-i}^{(x_\ell^{i-1})}\big) = e(g, \tilde{W}_{X\setminus S}).$$

If we further divide the above equality out of (11), we find that $\tilde{W} = \tilde{W}_{X\setminus S}/W_{X\setminus S}$ satisfies

$$\prod_{t:x_t\in X\setminus S} e(G_1, U_{tn})^{w_t} = e(g, \tilde{W}). \tag{13}$$

From (13), it is clear that $\tilde{W}$ is of the form

$$\tilde{W} = u^{\sum_{t:x_t\in X\setminus S} w_t\cdot(\alpha^{tn+1})} \cdot g_2^{t_2} \cdot g_3^{t_3}, \tag{14}$$

for some $t_2 \in \mathbb{Z}_{p_2}$ and $t_3 \in \mathbb{Z}_{p_3}$. The exponent of u in $\tilde{W}$ is thus a linear combination of the discrete logarithms of secret key components $U_{n+1}, U_{2n+1}, \ldots, U_{dn+1}$.

In the proof, we will distinguish two kinds of attacks against the collision-freeness property.

Type A attacks: are those for which $\tilde{W}$ lives in the subgroup $\mathbb{G}_{p_1p_3}$ (i.e., $t_2 = 0$).
Type B attacks: are such that $\tilde{W}$ has a $\mathbb{G}_{p_2}$ component (i.e., $t_2 \neq 0$).

The proof is organized as a hybrid argument over a sequence of $(d+1)n+1$ games with gradually varying distributions of PK and SK which are determined by a family of functions

$$\{F_\nu : [1,(d+1)n] \to \mathbb{Z}_{p_2}\}_{\nu=0}^{(d+1)n}$$

such that for all $j \in [1,(d+1)n]$,

$$F_\nu(j) = \begin{cases} 0 & \text{if } \nu = 0 \\ \sum_{i=1}^{\nu} r_j \cdot \alpha_i^j \bmod p_2 & \text{if } \nu \in [1,(d+1)n] \end{cases}$$

where $r_1, \ldots, r_{(d+1)n}, \alpha_1, \ldots, \alpha_{(d+1)n}$ are randomly distributed in $\mathbb{Z}_{p_2}$.

The sequence of games is – Game 0 (the real security game) followed by Game 1, Game 2, …, Game $(d+1)n$. In Game ν ($\nu \in [0, (d+1)n]$), the challenger provides the adversary with Type ν public keys, which are defined analogously to Type ν parameters in the proof of Theorem 1. In particular, Type 0 public keys refer to normal public keys, which are distributed as in the real scheme as $\{U_j\}_{j\in[1,(d+1)n]}$ do not contain a $\mathbb{G}_{p_2}$ component. In Type ν public keys, the group element U_j (for $j \in [1, (d+1)n]$,) has a $\mathbb{G}_{p_2}$ component determined by F_ν:

$$U_j = u^{\alpha^j} \cdot g_2^{F_\nu(j)} \cdot R_{3,j} \qquad \forall j,$$

thus defining a joint distribution on PK and $SK = \{U_{\ell n+1}\}_{\ell=1}^{d}$.

Using the function family $\{F_\nu\}_{\nu=1}^{(d+1)n}$, we can further classify Type B attacks into Type B.ν attacks for $1 \leq \nu \leq (d+1)n$:

Type B.ν attacks: are those where, in (14), the $\mathbb{G}_{p_2}$ component $g_2^{t_2}$ of $\tilde{W}$ is such that

$$t_2 = \sum_{t:x_t \in X\setminus S} w_t \cdot F_\nu(tn+1) \bmod p_2. \tag{15}$$

For notational convenience, we define Type B.0 attacks to be identical to Type A attacks.

As in the proof of Theorem 1, for each $\nu \in [1, (d+1)n]$, we call win_ν the event that the adversary $\mathcal{A}$ wins in Game ν. We also denote by E_ν the event that $\mathcal{A}$ mounts a Type B.ν attack when provided with a Type ν public key. We now describe the games and prove that, unless either Assumption 1 or Assumption 2 is false, $\mathcal{A}$ always mounts a Type B.ν attack when run on input of Type ν parameters. In Game $(d+1)n$, we also argue that the probability $\Pr[E_{(d+1)n}]$ can only be statistically negligible because $F_\nu(tn+1)$ is totally unpredictable and, since $w_t \neq 0 \bmod p_2$, relation (15) holds with negligible probability.

In Game 0, the public key PK provided to the adversary is generated as in the real scheme. We have $\Pr[\mathsf{win}_0] = \Pr[\mathsf{win}_0 \wedge E_0] + \Pr[\mathsf{win}_0 \wedge \neg E_0]$. The second term is bounded above by the advantage in breaking Assumption 1, as established by Lemma 3. Essentially, it says that a PPT adversary can only mount a Type A attack when fed with Type 0 parameters.

To bound the first term, Lemma 4 first implies that, if Assumption 2 holds, then $\Pr[\mathsf{win}_0 \wedge E_0]$ and $\Pr[\mathsf{win}_{(d+1)n} \wedge E_{(d+1)n}]$ are negligibly far apart. Lemma 4 provides evidence that the probability $\Pr[\mathsf{win}_\nu \wedge E_\nu]$ of $\mathcal{A}$ mounting an attack of the same type as the public key remains essentially the same throughout the entire sequence of games.

It only remains to show that $\Pr[\mathsf{win}_{(d+1)n} \wedge E_{(d+1)n}]$ is negligible. This is established in the proof of Lemma 5. Thus we have, $\Pr[\mathsf{win}_0] \leq \mathbf{Adv}^1_{\mathcal{B}}(\lambda) + (d+1)n \cdot \mathbf{Adv}^2_{\mathcal{B}}(\lambda) + \frac{1}{q_2}$, which is negligible under Assumption 1 and Assumption 2. □

Lemma 3. *If $\mathcal{A}$ is given a Type* 0 *public key and produces a Type B attack, then there exists an algorithm $\mathcal{B}$ such that* $\Pr[\mathsf{win}_0 \wedge \neg E_0] \leq \mathbf{Adv}^1_{\mathcal{B}}(\lambda)$*, where* $\mathbf{Adv}^1_{\mathcal{B}}(\lambda)$ *denotes $\mathcal{B}$'s advantage in breaking Assumption* 1.

Proof. Let $\mathcal{A}$ be an adversary that mounts a Type B attack when given a public key PK of Type A. We build an algorithm $\mathcal{B}$ that takes as input ($g \in \mathbb{G}_{p_1}, X_3 \in \mathbb{G}_{p_3}, T$) and finds an element η of $\mathbb{G}_{p_2p_3}$ with a non-trivial $\mathbb{G}_{p_2}$ component. In turn, such an $\eta \in \mathbb{G}_{p_2p_3}$

allows deciding whether $T \in \mathbb{G}_{p_1}$ or $T \in \mathbb{G}_{p_1 p_2}$ since $e(\eta, T) = 1_{\mathbb{G}_T}$ when $T \in \mathbb{G}_{p_1}$.

Algorithm $\mathcal{B}$ can faithfully generate PK using its input elements $g \in \mathbb{G}_{p_1}$ and $X_3 \in \mathbb{G}_{p_3}$. By hypothesis, $\mathcal{A}$ outputs set S and randomness γ that determine an accumulator value V as well as $X \not\subseteq S$ for which $\tilde{W} = \tilde{W}_{X \setminus S}/W_{X \setminus S}$ (as computed in the proof of Theorem 2) has a non-trivial $\mathbb{G}_{p_2}$ component. Moreover, from (14) se see that $\mathcal{B}$ can cancel out the $\mathbb{G}_{p_1}$ component of $\tilde{W}$ by computing $\eta = \tilde{W}/u^{\sum_{t:x_t \in X \setminus S} w_t \cdot (\alpha^{tn+1})}$, which indeed lives in $\mathbb{G}_{p_2 p_3}$ and has a non-trivial $\mathbb{G}_{p_2}$ component. At this point, $\mathcal{B}$ returns 1 (meaning that $T \in \mathbb{G}_{p_1}$) if $e(\eta, T) = 1_{\mathbb{G}_T}$ and 0 otherwise. □

Lemma 4. $|\Pr[\mathsf{win}_\nu \wedge E_\nu] - \Pr[\mathsf{win}_{\nu-1} \wedge E_{\nu-1}]| \leq \mathbf{Adv}^2_{\mathcal{B}}(\lambda)$*, where* $\mathbf{Adv}^2_{\mathcal{B}}(\lambda)$ *denotes the advantage of an algorithm* $\mathcal{B}$ *in breaking Assumption 2.*

Proof. Algorithm $\mathcal{B}$ inputs $(g, X_1X_2, Z_3, Y_2Y_3, T)$ and uses $\mathcal{A}$ to decide if $T \in_R \mathbb{G}_{p_1 p_3}$ or $T \in_R \mathbb{G}$. To this end, $\mathcal{B}$ generates the public key PK and the secret key SK as follows. It picks $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and defines

$$G_j = g^{(\alpha^j)} \qquad \forall j \in [1, n].$$

It also chooses $\alpha_1, \ldots, \alpha_{\nu-1}, r_1, \ldots, r_{\nu-1} \stackrel{R}{\leftarrow} \mathbb{Z}_N$ and computes

$$U_j = T^{(\alpha^j)} \cdot (Y_2Y_3)^{\sum_{i=1}^{\nu-1} r_j \cdot \alpha_i^j} \cdot R_{3,j} \qquad \forall j \in [1, (d+1)n],$$

for random $R_{3,j} \stackrel{R}{\leftarrow} \mathbb{G}_{p_3}$, so that u coincides with the $\mathbb{G}_{p_1}$ component of T. The public key

$$PK = \big(g, \{G_j\}_{j=1}^n, \{U_j\}_{j \in [1,(d+1)n] \setminus \{n+1, 2n+1, \ldots, dn+1\}}, Z_3\big)$$

is given to $\mathcal{A}$ while $\mathcal{B}$ keeps the secret key $SK := \{U_{n+1}, U_{2n+1}, \ldots, U_{dn+1}\}$. Then, $\mathcal{A}$ is expected to output

At this point, $\mathcal{B}$ computes $\tilde{W} = \tilde{W}_{X \setminus S}/W_{X \setminus S}$ as in the proof of Theorem 2. We know that $\tilde{W}$ is of the form

$$\tilde{W} = u^{\sum_{t:x_t \in X \setminus S} w_t \cdot (\alpha^{tn+1})} \cdot g_2^{t_2} \cdot g_3^{t_3},$$

where $u \in \mathbb{G}_{p_1}$ is the $\mathbb{G}_{p_1}$ component of T. From $\tilde{W}$, checks whether the equality

$$e\left(X_1X_2, \tilde{W}/\prod_{t:x_t \in X \setminus S} U_{tn+1}^{w_t}\right) = 1_{\mathbb{G}_T} \tag{16}$$

is satisfied. If so, it means that $\tilde{W}$ and $\prod_{t:x_t \in X \setminus S} U_{tn+1}^{w_t}$ are identical in their $\mathbb{G}_{p_1 p_2}$ component (and not only in their $\mathbb{G}_{p_1}$ component). It also means that the $\mathbb{G}_{p_2}$ component $g_2^{t_2}$ of $\tilde{W}$ satisfies the condition (15). If (16) holds, $\mathcal{B}$ deduces that $\mathcal{A}$'s attack and the resulting $\tilde{W}$ match the distribution of PK and outputs 1. Otherwise, it outputs 0.

We remark that, if $T \in_R \mathbb{G}_{p_1 p_2}$, then $\mathcal{B}$ is playing Game $\nu - 1$ with $\mathcal{A}$ since we have

$$U_j = u^{(\alpha^j)} \cdot Y_2^{\sum_{i=1}^{\nu-1} r_i \cdot \alpha_i^j} \cdot \tilde{R}_{3,j} \qquad \forall j \in [1, (d+1)n],$$

for some $\tilde{R}_{3,j} \in_R \mathbb{G}_{p_3}$ and relation (16) implies that event $\mathsf{win}_{\nu-1} \wedge E_{\nu-1}$ has occurred. If $T \in_R \mathbb{G}$, it can be written $T = u \cdot Y_2^{s_2} \cdot Y_3^{s_3}$, so that we have

$$U_j = u^{(\alpha^j)} \cdot Y_2^{s_2 \cdot \alpha^j + \sum_{i=1}^{\nu-1} r_i \cdot \alpha_i^j} \cdot \tilde{R}_{3,j} \qquad \forall j \in [1, (d+1)n],$$

with uniformly random $\tilde{R}_{3,j} \in_R \mathbb{G}_{p_3}$. In this case, $\mathcal{A}$'s view is identical to its view in Game ν, where $r_\nu = s_2 \bmod p_2$ and $\alpha_\nu = \alpha \bmod p_2$ (note that $\alpha \bmod p_2$ is uncorrelated to $\alpha \bmod p_1$) and relation (16) is equivalent to event $\mathsf{win}_\nu \wedge E_\nu$.

As a consequence, if moving from Game $\nu - 1$ to Game ν significantly modifies $\mathcal{A}$'s probability of mounting an attack of the same type as PK, so does it affect $\mathcal{B}$'s probability of outputting 1 when $T \in_R \mathbb{G}_{p_1 p_3}$ is replaced by $T \in_R \mathbb{G}$. □

Lemma 5. $\Pr[\mathsf{win}_{(d+1)n} \wedge E_{(d+1)n}] \leq 1/q_2$.

Proof. In the adversary's view, $F_{(d+1)n}(\cdot)$ is a random function. For this reason, the values $F_{(d+1)n}(n+1), F_{(d+1)n}(2n+1), \ldots, F_{(d+1)n}(dn+1)$ are independent and uniformly distributed over $\mathbb{Z}_{p_2}$ conditioned on

$$\{F_{(d+1)n}(j)\}_{j \in [1,(d+1)n] \setminus \{n+1, 2n+1, \ldots, dn+1\}}.$$

In the expression of $\tilde{W}$ in the proof of Theorem 2, to keep the exponent t_2 of g_2 consistent with the public key, the adversary has to predict

$$\sum_{t : x_t \in X \setminus S} w_t \cdot F_{(d+1)n}(tn + 1) \bmod p_2,$$

which is a linear combination of random function outputs. Since this combination has at least one non-zero coefficient $w_t \neq 0 \bmod p_2$, the probability of predicting the above (and thus the probability that (16) holds true) value is at most $1/p_2$. □