Candidate Trapdoor Claw-Free Functions from Group Actions with Applications to Quantum Protocols

Navid Alamati[1], Giulio Malavolta[2⋆], and Ahmadreza Rahimi[2]

[1] VISA Research, Palo Alto, CA, USA
nalamati@visa.com
[2] Max Planck Institute for Security and Privacy, Bochum, Germany
{giulio.malavolta,ahmadreza.rahimi}@mpi-sp.org

Abstract. Trapdoor Claw-free Functions (TCFs) are two-to-one trapdoor functions where it is computationally hard to find a claw, i.e., a colliding pair of inputs. TCFs have recently seen a surge of renewed interest due to new applications to quantum cryptography: as an example, TCFs enable a classical machine to verify that some quantum computation has been performed correctly. In this work, we propose a new family of (almost two-to-one) TCFs based on conjectured hard problems on isogeny-based group actions. This is the first candidate construction that is not based on lattice-related problems and the first scheme (from any plausible post-quantum assumption) with a *deterministic* evaluation algorithm. To demonstrate the usefulness of our construction, we show that our TCF family can be used to devise a *computational* test of a qubit, which is the basic building block used in the general verification of quantum computations.

1 Introduction

Trapdoor claw-free functions (TCFs) consist of pairs of functions $(f_0, f_1) : X \to Y$ that are easy to evaluate in the forward direction, but the knowledge of a trapdoor is required in order to efficiently invert such functions. Furthermore, for any y in the image of these two functions, there are exactly two pre-images (x_0, x_1) such that $f_0(x_0) = f_1(x_1) = y$ and the pair (x_0, x_1) is referred to as a *claw*. Claws are guaranteed to exist, though they are computationally hard to find, without the knowledge of the trapdoor. TCFs have been a central object in the theory of cryptography, and they have recently seen a surge of interest with a newly established connection with quantum cryptography. TCFs are the main cryptographic building block that enabled a series of recent breakthroughs in the area of quantum computation. To mention a few applications: the first protocol for testing the randomness of a single quantum device [BCM+18], classical verification of quantum computation [Mah18b], quantum fully homomorphic encryption [Mah18a], verifiable test of quantumness [BKVV20], remote state preparation [GV19], and deniable encryption [CGV22].

⋆ Research partially supported by the German Federal Ministry of Education and Research BMBF (grant 16K15K042, project 6GEM) and partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy - EXC 2092 CASA - 390781972.

At present, there is a *single* family of (noisy) TCFs [BCM+18] known to satisfy all of the properties needed for the above applications, whose security is based on the (quantum) hardness of the learning with errors (LWE) problem. While we have no reasons to cast doubts on the validity of this assumption, we believe that this situation is unsatisfactory and reflects our lack of understanding of cryptographic primitives useful for constructing protocols in the quantum regime.

This work aims to progress on this point and to place the security of the above protocols on broader cryptographic foundations. Towards this end, we turn our attention to alternative proposals for quantum-safe cryptographic schemes: Alongside lattices, another notable class of assumptions that enable advanced cryptographic applications (such as key exchange) is isogeny-based assumptions, including recent proposals based on group actions [CLM+18,BKV19]. Thus, we ask the following question:

Can we construct TCFs (or relaxations thereof) from isogeny-based group actions?

1.1 Our Results

We propose the first candidate construction of an "almost" TCF family from a class of isogeny-based assumptions, where by almost we mean that for all but an inverse polynomial fraction of inputs $x \in X$, there is an $x' \in X$ such that $f_0(x) = f_1(x')$. We later formalize this notion as a *weak TCF* (wTCF) family.

We show the security of our construction assuming an extended version of the linear hidden shift (LHS) problem (which plausibly holds over the isogeny-based group action of [BKV19]), introduced in [ADMP20]. A noteworthy aspect of our scheme is that the evaluation of the function is *deterministic*, which is in contrast with LWE-based schemes, where the function maps to a probability distribution. Thus, strictly speaking, our scheme is the first example of a wTCF *function* with plausible post-quantum security.

Our construction also satisfies a weaker variant of the *adaptive hardcore bit* property [BCM+18]: loosely speaking, it guarantees that one cannot simultaneously solve the adaptive hardcore bit problem for n independent instances, except with probability negligible in n. Interestingly, our proof strategy is completely different from that of [BCM+18], and does not rely on any leakage-resilience property. To obtain the stronger variant of the adaptive hardcore bit property (as formulated in [BCM+18]) we conjecture that computing the XOR of adaptive hardcore bits amplifies the security to negligibly close to $1/2$. In the context of one-wayness, it is known that direct-product hardness implies the XOR lemma [GNW11,GSV18], and we leave open the problem of proving a similar statement for the adaptive hardcore bit property.

To substantiate the usefulness of our construction, we show that our wTCF family can be used to devise a *computational* test of qubit [BCM+18,Vid20], which is the basic building block used in the general verification of quantum computations.

1.2 Technical Overview

We now provide a simplified overview of how we construct a wTCF family from an assumption that plausibly holds over isogeny-based group actions. We present our

overview entirely in terms of group actions (based on the framework of [ADMP20]), and thus we do not assume any familiarity with CSIDH and its variants [CLM+18,BKV19]. The starting point for our construction is a recently introduced assumption in [ADMP20], called the linear hidden shift (LHS) assumption. In a nutshell, LHS assumption over a regular and abelian group action $\star : \mathbb{G} \times \mathbb{X} \to \mathbb{X}$ states that for any $\ell = \mathrm{poly}(\lambda)$, if $\mathbf{M} \leftarrow \mathbb{G}^{\ell \times n}$, $\mathbf{v} \leftarrow \{0,1\}^n$, and $\mathbf{x} \leftarrow \mathbb{X}^\ell$ (for some sufficiently large n) then

$$(\mathbf{x}, \mathbf{M}, \mathbf{M}\mathbf{v} \star \mathbf{x}) \overset{c}{\approx} (\mathbf{x}, \mathbf{M}, \mathbf{u}),$$

where $\mathbf{u} \leftarrow \mathbb{X}^\ell$ is sampled uniformly and $\star$ is applied component-wise. Given this assumption, we rely on an observation by [KCVY21] to construct a function family that is (almost) 2-to-1. It can be verified by inspection that if $B = \mathrm{poly}(\lambda)$ is a large enough integer, then for any injective function $\bar{f}$ whose domain is a superset of $[B+1]^n$, the function f with domain $\{0,1\} \times [B]^n$ defined by $f(b \in \{0,1\}, \mathbf{s} \in [B]^n) = \bar{f}(\mathbf{s} + b \cdot \mathbf{v})$ is an "almost" 2-to-1 function. Based on this simple observation, an initial attempt to define a claw-free "almost" 2-to-1 function (from LHS) would be

$$f_{\mathsf{pp}}(b, \mathbf{s}) = \mathbf{M}(\mathbf{s} + b \cdot \mathbf{v}) \star \mathbf{x}, \quad \mathsf{pp} = (\mathbf{M} \leftarrow \mathbb{G}^{n \times n}, \mathbf{M}\mathbf{v} \star \mathbf{x}, \mathbf{x} \leftarrow \mathbb{X}^n), \qquad (*)$$

where $\mathbf{v} \leftarrow \{0,1\}^n$. As a sanity check, any claw-pair $((0, \mathbf{s}_0), (1, \mathbf{s}_1))$ can be used to break the LHS assumption by simply computing $\mathbf{v} = \mathbf{s}_0 - \mathbf{s}_1$. There are two major issues with the initial attempt above: (1) unlike the DDH-based construction of [KCVY21], a cryptographic group action does not seem to be amenable for a "DDH-style" trapdoor [FGK+10] (in fact, any such technique would immediately break the post-quantum security of LHS assumption), and (2) it is not clear how to translate the LWE-based proof of adaptive hardcore bit property from [BCM+18] to the group action setting. Indeed, the latter seems to be a major bottleneck, because [BCM+18] relies on the lossy mode of LWE to prove the adaptive hardcore bit property via a lossiness argument, a technique that seems to be out of reach based on our current understanding of cryptographic group actions. At a high level, any change in the structure of matrix $\mathbf{M}$ (say using a "rank" 1 matrix) can be easily detected by a quantum adversary. Thus, we opt for an entirely *computational* approach to prove the adaptive hardcore bit property. We first describe our approach for showing adaptive hardcore bit property, and later we explain how to add input recoverability based on a related computational assumption.

From a Claw-Based Inner Product to a Shift-Based Equation. Note that for the function (family) f_{pp} above $(*)$, the adaptive hardcore bit property means that no QPT adversary can simultaneously hold a preimage $(b, \mathbf{s}_b)$ and a pair $(\mathbf{d}, c \in \{0,1\})$ such that $c = \langle \mathbf{d}, \mathbf{s}_0 \oplus \mathbf{s}_1 \rangle$, where $\mathbf{s}_{1-b}$ is the preimage of $f_{\mathsf{pp}}(b, \mathbf{s}_b)$ such that $\mathbf{s}_{1-b} \neq \mathbf{s}_b$ and the inner product is computed over $\mathbb{F}_2$. To simplify the proof, an observation by [BCM+18] showed that any such tuple $(b, \mathbf{s}_b, \mathbf{d}, c)$ can be transformed into a binary equation in terms of the *shift vector* $\mathbf{v}$, i.e., there is an efficient transformation T that given $(b, \mathbf{s}_b, \mathbf{d}, c)$ outputs a binary vector $\mathbf{d}'$ and $c' \in \{0,1\}$ such that $c' = \langle \mathbf{d}', \mathbf{v} \rangle$, and that for a uniformly chosen $\mathbf{d}$ the resulting $\mathbf{d}'$ is non-zero with overwhelming probability. Thus, the adaptive hardcore bit property can be rephrased as the infeasibility of computing *any* non-trivial parity of the shift vector $\mathbf{v}$ with a probability noticeably

more than 1/2. Although our final construction will be quite different from the simple one outlined above ($*$), it would still be amenable to a transformation from a claw-based inner product into a shift-based equation. Therefore, we focus on the latter in the remaining part of this overview. Looking ahead, in our final construction the shift vector $\mathbf{v}$ will consist of n binary vectors $\mathbf{v}_i$ (for $i \in [n]$). In the next step, we describe a generic approach to prove that no attacker can succeed in outputting n non-zero vectors $\mathbf{d}'_i$ and n bits c_i ($i \in [n]$) such that $c_i = \langle \mathbf{d}'_i, \mathbf{v}_i \rangle$ for all $i \in [n]$.

Direct-Product Adaptive Hardcore Bit. Let $F_{\mathsf{pp}} : \{0,1\}^n \to Y$ be a function (family) such that pp is generated via a randomized algorithm Gen. In addition, assume that F satisfies *correlated pseudorandomness*, i.e., for uniformly sampled $\mathbf{w} \leftarrow \{0,1\}^n$ and n independently sampled $(\mathsf{pp}_i)_{i\in[n]}$ (via Gen) we have

$$(\mathsf{pp}_1, \ldots, \mathsf{pp}_n, F_{\mathsf{pp}_1}(\mathbf{w}), \ldots, F_{\mathsf{pp}_n}(\mathbf{w})) \overset{c}{\approx} (\mathsf{pp}_1, \ldots, \mathsf{pp}_n, u_1, \ldots, u_n),$$

where $u_i \leftarrow Y$ for $i \in [n]$. Suppose that there is a procedure $\mathcal{P}$ that given $(\mathsf{pp}_i, F_{\mathsf{pp}_i}(\mathbf{w}))_{i\in[n]}$ (where pp_i is generated independently for $i \in [n]$) and n (random) binary vectors $(\mathbf{r}_i)_{i\in[n]}$, it outputs

$$(\mathsf{pp}'_i, F_{\mathsf{pp}'_i}(\mathbf{w} \oplus \mathbf{r}_i))_{i\in[n]}$$

such that

$$(\mathsf{pp}'_i, F_{\mathsf{pp}'_i}(\mathbf{w} \oplus \mathbf{r}_i))_{i\in[n]} \overset{s}{\approx} (\overline{\mathsf{pp}}_i, F_{\overline{\mathsf{pp}}_i}(\mathbf{v}_i))_{i\in[n]}, \qquad (**)$$

where $\mathbf{v}_i \leftarrow \{0,1\}^n$ for $i \in [n]$ and each $\overline{\mathsf{pp}}_i$ is generated independently. Moreover, the procedure $\mathcal{P}$ should map a random tuple $(\mathsf{pp}_i, u_i)_{i\in[n]}$ (where $u_i \leftarrow Y$) to a random tuple.

Given such a function family with corresponding procedure $\mathcal{P}$, below we briefly outline a reduction that shows for any QPT[1] adversary $\mathcal{A}$, given $(\mathsf{pp}_i, F_{\mathsf{pp}_i}(\mathbf{v}_i))_{i\in[n]}$ where pp_i and $\mathbf{v}_i$ are sampled independently for $i \in [n]$, it is infeasible to produce n non-zero vectors $\mathbf{d}'_i$ and n bits c_i (for $i \in [n]$) such that $c_i = \langle \mathbf{d}'_i, \mathbf{v}_i \rangle$ for all $i \in [n]$, where the inner product is computed over $\mathbb{F}_2$. We informally refer to this property as *direct-product* adaptive hardcore bit property.

Let $H = (\mathsf{pp}_1, \ldots, \mathsf{pp}_n, y_1, \ldots, y_n)$ be a correlated pseudorandomness challenge. The reduction samples $\mathbf{r}_i$ for $i \in [n]$ and it runs $\mathcal{P}$ on $(H, \mathbf{r}_1, \ldots, \mathbf{r}_n)$. Let $(\mathbf{d}'_i, \beta_i)$ for $i \in [n]$ be the output of $\mathcal{A}$. Observe that if the advantage of $\mathcal{A}$ is non-negligible and H is pseudorandom, i.e., $y_i = F_{\mathsf{pp}_i}(\mathbf{w})$ for all $i \in [n]$, the reduction can use $(\mathbf{d}'_i, \beta_i)$ and $\mathbf{r}_i$ to compute $c_i = \langle \mathbf{d}'_i, \mathbf{w} \rangle$ for $i \in [n]$. If there exists an index n' such that $\mathbf{d}'_{n'}$ lies in the span of $(\mathbf{d}'_1, \ldots, \mathbf{d}'_{n'-1})$, i.e.,

$$\mathbf{d}'_{n'} = \sum_{i=1}^{n'-1} \alpha_i \mathbf{d}'_i, \quad (\alpha_1, \ldots, \alpha_{n'-1}) \in \{0,1\}^{n'-1},$$

the reduction can simply check $c_{n'} \overset{?}{=} \sum_{i=1}^{n'-1} \alpha_i c_i$. If the equality holds the reduction outputs 0, otherwise it outputs a random bit. On the other hand, a routine information-theoretic argument shows that if H is a truly random tuple then the check above passes

[1] The reduction is entirely classical, so if correlated pseudorandomness holds with respect to all classical PPT adversaries, then the proposition holds for the same class of adversaries as well.

with a probability close to 1/2, because $\mathbf{r}_i$ (for any $i \in [n]$) is statistically hidden from the view of $\mathcal{A}$, allowing us to deduce the direct-product adaptive hardcore bit property (a slight modification of the argument also works in case all $\mathbf{d}'_i$ for $i \in [n]$ are linearly independent).

So far, we argued that if $F_{\mathsf{pp}} : \{0,1\}^n \to Y$ is a function family with correlated pseudorandomness and a corresponding procedure $\mathcal{P}$, it also satisfies the direct-product adaptive hardcore bit property. In the next step, we rely on a conjecture to deduce the (plain) adaptive hardcore bit property (defined below), which will allow us to deduce the adaptive hardcore bit property (for an almost 2-to-1 function) in our final construction. A non-adaptive version of the following conjecture has already been proved via a transformation from direct-product hardness to (Yao's) XOR lemma [GNW11].

Conjecture 1 (Informal). If $F_{\mathsf{pp}} : \{0,1\}^n \to Y$ is a function family (with the properties described above) that satisfies the direct-product adaptive hardcore bit property, it also satisfies the following adaptive hardcore bit property defined as:

$$\Pr\left[\mathcal{A}\big(\{\mathsf{pp}_i\}_{i\in[n]}, \{F_{\mathsf{pp}_i}(\mathbf{v}_i)\}_{i\in[n]}\big) \to \left(\{\mathbf{d}'_i \neq 0^n\}_{i\in[n]}, \bigoplus_{i=1}^{n}\langle \mathbf{d}'_i, \mathbf{v}_i\rangle\right)\right] \leq 1/2+\mathrm{negl}\,.$$

Remark 1. While the adaptive hardcore bit property in the conjecture above is different from the adaptive hardcore bit property in the case of the (2-to-1) TCF family, they can be related via the transformation that has been described before, namely the transformation from a claw-based inner product to a shift-based equation.

Realizing (Direct-Product) Adaptive Hardcore Bit. It remains to show how we can realize the abstraction above using LHS or a related assumption. First, observe that correlated pseudorandomness can be easily handled since for n randomly generated pp_i of the following form, it follows immediately by the LHS assumption that for $i \in [n]$:

$$(\mathbf{x}_i, \mathbf{M}_i, \mathbf{M}_i\mathbf{w} \star \mathbf{x}_i)_{i\in[n]} \overset{c}{\approx} (\mathbf{x}_i, \mathbf{M}_i, \mathbf{u}_i)_{i\in[n]} \quad \mathsf{pp}_i = (\mathbf{M}_i \leftarrow \mathbb{G}^{n\times n}, \mathbf{x}_i \leftarrow \mathbb{X}^n)$$

where $\mathbf{w} \leftarrow \{0,1\}^n$ and $\mathbf{u}_i \leftarrow \mathbb{X}^n$ for $i \in [n]$. However, it is unclear how to find a corresponding efficiently computable procedure $\mathcal{P}$ (defined in the previous part). To get around this issue, we work with a slightly different form of the LHS assumption. Specifically, we can work with the following form of the LHS assumption (which is implied by the original LHS assumption via a simple reduction):

$$\mathsf{pp}_i = (\mathbf{M}_i^{(0)} \leftarrow \mathbb{G}^{n\times n}, \mathbf{M}_i^{(1)} \leftarrow \mathbb{G}^{n\times n}, \mathbf{x}_i \leftarrow \mathbb{X}^n), \quad i \in [n],$$

$$\big(\mathbf{x}_i, \mathbf{M}_i^{(0)}, \mathbf{M}_i^{(1)}, \big[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{w}) + \mathbf{M}_i^{(1)}\mathbf{w}\big] \star \mathbf{x}_i\big)_{i\in[n]} \overset{c}{\approx} \big(\mathbf{x}_i, \mathbf{M}_i^{(0)}, \mathbf{M}_i^{(1)}, \mathbf{u}_i\big)_{i\in[n]},$$

where $\mathbf{1}$ is an all-one vector. It is not hard to see that based on the new form of the assumption, given $(\mathsf{pp}_i, F_{\mathsf{pp}_i}(\mathbf{w}))$ and binary vectors $\mathbf{r}_i$ for $i \in [n]$, one can efficiently produce

$$(\mathsf{pp}'_i, F_{\mathsf{pp}'_i}(\mathbf{w} \oplus \mathbf{r}_i)), \quad i \in [n],$$

where $F_{\mathsf{pp}'_i}(\mathbf{w}\oplus\mathbf{r}_i) = F_{\mathsf{pp}_i}(\mathbf{w})$, and pp'_i is simply obtained by swapping the jth column of $\mathbf{M}_i^{(0)}$ and $\mathbf{M}_i^{(1)}$ for all positions j such that the jth bit of $\mathbf{r}_i$ is 1. One can also verify that the aforementioned procedure also satisfies the indistinguishability $(**)$.

Input Recoverability and Extended LHS Assumption. To add input recoverability, we informally define one-matrix version of an extended form of the LHS assumption, which asserts that

$$\big(\mathbf{M}, \mathbf{m}, \mathbf{x}^{(\beta)}, \mathbf{y}^{(\beta)}\big)_{\beta\in\{0,1\}} \overset{c}{\approx} \big(\mathbf{M}, \mathbf{m}, \mathbf{u}^{(\beta)}, \mathbf{u'}^{(\beta)}\big)_{\beta\in\{0,1\}},$$

where each of the terms above is distributed as

$$\mathbf{w} \leftarrow \{0,1\}^n, \quad \mathbf{M} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m} \leftarrow \mathbb{G}^n, \quad \mathbf{x}^{(0)} \leftarrow \mathbb{X}^n,$$
$$\mathbf{t} \leftarrow \mathbb{G}^n, \quad \mathbf{u}^{(\beta)} \leftarrow \mathbb{X}^n, \quad \mathbf{u'}^{(\beta)} \leftarrow \mathbb{X}^n, \quad (\beta \in \{0,1\})$$

$$\mathbf{x}^{(1)} := \big[\mathbf{M}\mathbf{w}\big] \star \mathbf{x}^{(0)}, \quad \mathbf{y}^{(0)} := \mathbf{t} \star \mathbf{x}^{(0)},$$
$$\mathbf{y}^{(1)} := \big[\mathbf{M}\mathbf{w} + \mathbf{m} \odot \mathbf{w}\big] \star \mathbf{y}^{(0)},$$

and $\odot$ denotes the component-wise product of an integer and a group element (defined in a natural way). Note that for the left-hand side of the assumption above, knowing (a trapdoor) $\mathbf{t}$ is enough to recover $\mathbf{w}$,[1] since

$$-\mathbf{t} \star \mathbf{y}^{(1)} = (\mathbf{m} \odot \mathbf{w}) \star \mathbf{x}^{(1)}.$$

Final Construction. Now we provide the final construction of our wTCF family. To generate a key-trapdoor pair, for each $i \in [n]$ and $\beta \in \{0,1\}$ sample

$$\mathbf{v}_i \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n,$$

and set

$$\mathbf{x}_i^{(1)} := \big[\mathbf{M}_i^{(0)}(\mathbf{1} - \mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i\big] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \big[\mathbf{M}_i^{(0)}(\mathbf{1} - \mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i + \mathbf{m}_i^{(0)} \odot (\mathbf{1} - \mathbf{v}_i) + \mathbf{m}_i^{(1)} \odot \mathbf{v}_i\big] \star \mathbf{y}_i^{(0)},$$

where $\odot$ denotes component-wise product. Output $(\mathsf{ek}, \mathsf{td})$ where

$$\mathsf{td} = \big(\mathbf{v}_i, \mathbf{t}_i\big)_{i\in[n]}, \quad \mathsf{ek} = \big(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\big)_{i\in[n],\beta\in\{0,1\}}.$$

To evaluate the function $f_{\mathsf{ek},b}$ on input $(\mathbf{s}_i)_{i\in[n]} \in ([B]^n)^n$, output $(\bar{\mathbf{z}}_i, \mathbf{z}_i)$ for $i \in [n]$ where

$$\bar{\mathbf{z}}_i = \big[(1-b)\cdot\mathbf{M}_i^{(0)}\mathbf{1} + \big(\mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}\big)\mathbf{s}_i\big] \star \mathbf{x}_i^{(b)},$$
$$\mathbf{z}_i = \big[(1-b)\cdot\mathbf{M}_i^{(0)}\mathbf{1} + \big(\mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}\big)\mathbf{s}_i + (1-b)\cdot\mathbf{m}_i^{(0)} + (\mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}) \odot \mathbf{s}_i\big] \star \mathbf{y}_i^{(b)}.$$

Observe that if $f_{\mathsf{ek},b}((\mathbf{s}_i)_{i\in[n]}) = (\bar{\mathbf{z}}_i, \mathbf{z}_i)_{i\in[n]}$ then the following relation holds for any $i \in [n]$:

$$(-\mathbf{t}_i - \mathbf{m}_i^{(0)}) \star \mathbf{z}_i = \big[(\mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}) \odot (\mathbf{s}_i + b\cdot\mathbf{v}_i)\big] \star \bar{\mathbf{z}}_i.$$

Because the action is applied component-wise and each entry of $\mathbf{s}_i$ lies in $[B]$, one can recover each entry of $\mathbf{s}_i$ efficiently by a simple brute force, since both $\mathbf{v}_i$ and $\mathbf{t}_i$ are included in the trapdoor.

[1] Note that knowledge of $\mathbf{t}$ is enough to recover $\mathbf{w}$ even if $\mathbf{w}$ is non-binary but with short entries, i.e., if each entry of $\mathbf{w}$ is polynomially bounded.

Computational Qubit Test. To exemplify the usefulness of our wTCF family, we show how it can be used as the cryptographic building block in the computational qubit test described by Vidick [Vid20]. Such a test allows a quantum prover to certify the possession of a qubit in its internal state. Importantly, the verifier and the communication are entirely classical. The protocol that we present is largely unchanged from [Vid20], except for a few syntactical modifications due to the presence of non-perfect matchings in the input domain of our wTCFs. For more details, we refer the reader to Section 5. We view this protocol as a promising first step towards the usage of our isogeny-based wTCF in more complex protocols for the verification of more involved quantum tasks.

2 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter. A function negl is negligible if it vanishes faster than any polynomial. We denote by $[n]$ the set $\{1, \ldots, n\}$.

2.1 Quantum Information

We recall a few facts about quantum information to establish some notation and we refer the reader to [NC02] for a more comprehensive overview. A (pure) quantum state $|\psi\rangle$ is a unit vector in a separable Hilbert space $\mathcal{H}$. Throughout this work, we will only consider finite-dimensional Hilbert spaces and so we will always assume that $\mathcal{H} \simeq \mathbb{C}^d$, for some integer $d \geq 1$. A Projector-Valued Measure (PVM) consists of a set of projectors $\{\Pi_i\}$ that sum up to identity, and if Π_i are not required to be projectors, it is called a Positive Operator-Valued Measure (POVM). Given a POVM $\{\Pi_i\}$, the *Born rule* establishes that measuring a state $|\psi\rangle$ will yield outcome i with probability $\langle\psi| \Pi_i |\psi\rangle$.

An *observable* O is a Hermitian operator on $\mathcal{H}$. Let $O = \sum_i \lambda_i \Pi_i$ be the spectral decomposition of O, then we call an *eigenstate* of O a pure state $|\psi\rangle$ such that $\Pi_i |\psi\rangle$ will deterministically yield outcome λ_i, when measured according to O. Throughout this work, we will only consider *binary observables* O such that $O^2 = \mathsf{Id}$, and that $O = \Pi_0 - \Pi_1$. I.e., they are the sum of two projectors and have eigenvalues $\lambda_i \in \{-1, +1\}$. It is convenient to define the *expected outcome* of an observable O on a state $|\psi\rangle$ as

$$\sum_i \lambda_i \langle\psi| \Pi_i |\psi\rangle = \langle\psi| O |\psi\rangle .$$

2.2 Cryptographic Group Actions and Extended LHS Assumption

In this part we recall some definitions related to cryptographic group actions from [ADMP20], which provided a framework to capture certain isogeny-based assumptions (e.g., variants of CSIDH [CLM+18,BKV19]). We refer to [ADMP20] for a detailed explanation of these definitions. Towards the end of the section, we provide a definition of extended linear hidden shift assumption, from which we later show the construction of wTCF family. We present our results entirely in terms of group actions with certain hardness

properties (based on the framework of [ADMP20]), and thus we do not assume familiarity with CSIDH and its variants [CLM+18,BKV19]. We refer to [Pei20,BS20] for an overview of quantum attacks against CSIDH for certain choices of parameters.

Throughout the paper, we use the abbreviated notation $(\mathbb{G}, \mathbb{X}, \star)$ to denote a group action $\star : \mathbb{G} \times \mathbb{X} \to \mathbb{X}$. Moreover, we are going to assume that group actions are abelian and *regular*, i.e., both free and transitive (which is the case for all isogeny-based group actions). For such group actions, we have $|\mathbb{G}| = |\mathbb{X}|$. Note that if a group action is regular, then for any $x \in \mathbb{X}$, the map $f_x : g \mapsto g \star x$ defines a bijection between $\mathbb{G}$ and $\mathbb{X}$.

We recall the definition of an effective group action (EGA) from [ADMP20]. In a nutshell, an effective group action allows us to efficiently perform certain tasks over $\mathbb{G}$ (e.g., group operation, inversion, and sampling uniformly) efficiently, along with an efficient procedure to compute the action of any group element on any set element. As a concrete example, a variant of CSIDH [BKV19] (called "CSI-FiSh") can be modeled as an effective group action, for which the group $\mathbb{G}$ is isomorphic to $(\mathbb{Z}_N, +)$.[1]

Definition 1 (Effective Group Action (EGA)). *A group action* $(\mathbb{G}, \mathbb{X}, \star)$ *is* effective *if it satisfies the following properties:*

1. *The group* $\mathbb{G}$ *is finite and there exist efficient (PPT) algorithms for:*
   (a) *Membership testing (deciding whether a binary string represents a group element).*
   (b) *Equality testing and sampling uniformly in* $\mathbb{G}$.
   (c) *Group operation and computing inverse of any element.*
2. *The set* $\mathbb{X}$ *is finite and there exist efficient algorithms for:*
   (a) *Membership testing (to check if a string represents a valid set element),*
   (b) *Unique representation.*
3. *There exists a distinguished element* $x_0 \in \mathbb{X}$ *with known representation.*
4. *There exists an efficient algorithm that given any* $g \in \mathbb{G}$ *and any* $x \in \mathbb{X}$*, outputs* $g \star x$.

Notation. For a group action $\star : \mathbb{G} \times \mathbb{X} \to \mathbb{X}$, we always use the additive notation $+$ to denote the group operation in $\mathbb{G}$. Since $\mathbb{G}$ is abelian, it can be viewed as a $\mathbb{Z}$-module, and hence for any $z \in \mathbb{Z}$ and $g \in \mathbb{G}$ the term zg is well-defined. This property naturally extends to vectors and matrices as well, so if $\mathbf{g} \in G^n$ and $\mathbf{z} \in \mathbb{Z}^n$ for some $n \in \mathbb{N}$, then we use $\langle \mathbf{g}, \mathbf{z} \rangle$ to denote $\sum_{i=1}^{n} z_i g_i$. Thus, for any matrix $\mathbf{M} \in \mathbb{G}^{m \times n}$ and any vector $\mathbf{z} \in \mathbb{Z}^n$, the term $\mathbf{Mz}$ is also well-defined.

For any two vectors $\mathbf{z} \in \mathbb{Z}^n$ and $\mathbf{g} \in \mathbb{G}^n$ we use the notation $\mathbf{z} \odot \mathbf{g}$ to denote a vector whose ith component is $z_i g_i$ (component-wise/Hadamard product). The group action also extends to the direct product group $\mathbb{G}^n$ for any positive integer n. If $\mathbf{g} \in \mathbb{G}^n$ and $\mathbf{x} \in X^n$, we use $\mathbf{g} \star \mathbf{x}$ to denote a vector of set elements whose ith component is $g_i \star x_i$.

[1] Although we present our results in terms of EGA, one can also obtain the same results from a *restricted* EGA assuming a one-time quantum preprocessing, since EGA and restricted EGA are quantumly equivalent [ADMP20].

Definition 2 (Weak Pseudorandom EGA). *An (effective) group action $(\mathbb{G}, \mathbb{X}, \star)$ is said to be a weak pseudorandom EGA if it holds that*

$$(x, y, t \star x, t \star y) \overset{c}{\approx} (x, y, u, u'),$$

where $x \leftarrow \mathbb{X}$, $y \leftarrow \mathbb{X}$, $t \leftarrow \mathbb{G}$, $u \leftarrow \mathbb{X}$, and $u' \leftarrow \mathbb{X}$.

Definition 3 (Linear Hidden Shift (LHS) assumption [ADMP20]). *Let $(\mathbb{G}, \mathbb{X}, \star)$ be an effective group action (EGA), and let $n > \log|\mathbb{G}| + \omega(\log\lambda)$ be a positive integer. We say that liner hidden shift (LHS) assumption holds over $(\mathbb{G}, \mathbb{X}, \star)$ if for any $\ell = \mathrm{poly}(\lambda)$ the following holds:*

$$(\mathbf{x}, \mathbf{M}, \mathbf{M}\mathbf{w} \star \mathbf{x}) \overset{c}{\approx} (\mathbf{x}, \mathbf{M}, \mathbf{u}),$$

where each of the terms above is distributed as

$$\mathbf{x} \leftarrow \mathbb{X}^{\ell}, \quad \mathbf{M} \leftarrow \mathbb{G}^{\ell \times n}, \quad \mathbf{w} \leftarrow \{0,1\}^{n}, \quad \mathbf{u} \leftarrow \mathbb{X}^{\ell}.$$

Definition 4 (Extended LHS assumption). *Let $(\mathbb{G}, \mathbb{X}, \star)$ be an effective group action (EGA), and let $n > \log|\mathbb{G}| + \omega(\log\lambda)$ be a positive integer. We say that extended LHS assumption holds over $(\mathbb{G}, \mathbb{X}, \star)$ if for any $\ell = \mathrm{poly}(\lambda)$ the following holds:*

$$\left(\mathbf{M}_i, \mathbf{m}_i, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[\ell],\beta\in\{0,1\}} \overset{c}{\approx} \left(\mathbf{M}_i, \mathbf{m}_i, \mathbf{u}_i^{(\beta)}, \mathbf{u'}_i^{(\beta)}\right)_{i\in[\ell],\beta\in\{0,1\}},$$

where each of the terms above is distributed as

$$\mathbf{w} \leftarrow \{0,1\}^{n}, \quad \mathbf{M}_i \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i \leftarrow \mathbb{G}^{n}, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^{n},$$
$$\mathbf{t}_i \leftarrow \mathbb{G}^{n}, \quad \mathbf{u}_i^{(\beta)} \leftarrow \mathbb{X}^{n}, \quad \mathbf{u'}_i^{(\beta)} \leftarrow \mathbb{X}^{n},$$

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M}_i\mathbf{w}\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \left[\mathbf{M}_i\mathbf{w} + \mathbf{m}_i \odot \mathbf{w}\right] \star \mathbf{y}_i^{(0)}.$$

Remark 2. Note that in the assumption above if $\mathbf{y}_i^{(1)}$ were distributed as $\mathbf{y}_i^{(1)} = \left[\mathbf{M}_i\mathbf{w}\right] \star \mathbf{y}_i^{(0)}$, then the extended LHS assumption would be implied by any weak pseudorandom EGA over which LHS assumption holds. In other words, the presence of the term $\mathbf{m}_i \odot \mathbf{w}$ makes the extended LHS assumption seemingly stronger than the plain LHS assumption.

3 Weak Trapdoor Claw-Free Functions

We define the notion of a weak trapdoor claw-free function (wTCF) family. We adopt a slightly simplified syntax compared to [BCM+18] as each function in our definition of wTCF family will be a deterministic function rather than mapping to a probability distribution.

Definition 5 (wTCF). *Let $n = n(\lambda)$ be an integer such that $n = \mathrm{poly}(\lambda)$. Let $\mathcal{F}$ be a family of functions*

$$\mathcal{F} = \{f_{\mathsf{ek},b} : X^n \to Y\}_{(\mathsf{ek},b)\in K\times\{0,1\}},$$

where X, Y, and K are finite sets indexed by λ, and K denotes the the key space. We say that $\mathcal{F}$ is a weak trapdoor claw-free (wTCF) function family if it satisfies the following properties:

1. *There exists a PPT algorithm* Gen *which generates an evaluation key* ek *along with a trapdoor* td *as* $(\mathsf{ek},\mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$.
2. *For all but a negligible fraction of key-trapdoor pairs* $(\mathsf{ek},\mathsf{td}) \in \mathit{supp}(\mathsf{Gen}(1^\lambda))$, *the following properties hold.*
   (a) *There exists an efficient algorithm* Invert *that for any $b \in \{0,1\}$ and any $\mathbf{x} \in X^n$, it holds that*
   $$\mathsf{Invert}(\mathsf{td}, b, f_{\mathsf{ek},b}(\mathbf{x})) = \mathbf{x}.$$
   (b) *There exists two* dense *subsets $\mathbf{X}_0 \subseteq X^n$ and $\mathbf{X}_1 \subseteq X^n$ and a perfect matching $R_{\mathsf{ek}} \subseteq \mathbf{X}_0 \times \mathbf{X}_1$ such that for any $(\mathbf{x}_0,\mathbf{x}_1) \in \mathbf{X}_0 \times \mathbf{X}_1$ it holds that $f_{\mathsf{ek},0}(\mathbf{x}_0) = f_{\mathsf{ek},1}(\mathbf{x}_1)$ iff $(\mathbf{x}_0,\mathbf{x}_1) \in R_{\mathsf{ek}}$, where a dense subset $\mathbf{X} \subseteq X^n$ is defined as a subset that satisfies*
   $$\Pr_{\mathbf{x}\leftarrow X^n}[\mathbf{x} \in \mathbf{X}] \geq 1 - n^{-c},$$
   for some constant $c \geq 1$. For any $\mathbf{x} \in X^n$, membership in $\mathbf{X}_0$ or $\mathbf{X}_1$ can be checked efficiently given the trapdoor td. *In addition, there exists a dense subset $\bar{\mathbf{X}} \subseteq \mathbf{X}_0 \cap \mathbf{X}_1 \subseteq X^n$ such that membership in $\bar{\mathbf{X}}$ can be checked without* td.
   Informally, this property means that a randomly sampled $\mathbf{x} \leftarrow X^n$ lies in $\bar{\mathbf{X}} \subseteq \mathbf{X}_0 \cap \mathbf{X}_1$ with "good" probability. Moreover, for any $\mathbf{x} \in \mathbf{X}_0 \cap \mathbf{X}_1$ and any $b \in \{0,1\}$, the image $y = f_{\mathsf{ek},b}(\mathbf{x})$ has exactly one preimage $\mathbf{x}_0 \in \mathbf{X}_0$ under $f_{\mathsf{ek},0}$ and one preimage $\mathbf{x}_1 \in \mathbf{X}_1$ under $f_{\mathsf{ek},1}$.
3. (a) *There exists an efficiently computable "binary encoding" function $\mathsf{B} : X^n \to \{0,1\}^{n\ell}$ such that B^{-1} is also efficiently computable on the range of* B.
   (b) *For any $b \in \{0,1\}$ and any $\mathbf{x} \in X^n$, there exists a set $\mathbb{Y}_{b,\mathbf{x}} \subseteq \{0,1\}^{n\ell}$ such that*
   $$\Pr_{\mathbf{d}\leftarrow\{0,1\}^{n\ell}}[\mathbf{d} \notin \mathbb{Y}_{b,\mathbf{x}}] \leq \mathrm{negl},$$
   and membership in $\mathbb{Y}_{b,\mathbf{x}}$ can be checked efficiently given b and $\mathbf{x}$.
   (c) *Let W_{ek} be a (key-dependent) set of tuples defined as*
   $$W_{\mathsf{ek}} = \left\{ \left(b, \mathbf{x}_b, \mathbf{d}, \left(\langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1)\rangle\right)_{i\in[n]}\right) \middle| \begin{array}{l} b \in \{0,1\}, (\mathbf{x}_0,\mathbf{x}_1) \in R_{\mathsf{ek}}, \\ \mathbf{d} \in \mathbb{Y}_{0,\mathbf{x}_0} \cap \mathbb{Y}_{1,\mathbf{x}_1} \end{array} \right\},$$
   where $\mathbf{d}_i$ and $\mathsf{B}_i(\cdot)$ denote the ith ℓ-bit chunk of $\mathbf{d}$ and $\mathsf{B}(\cdot)$, respectively (the inner product is computed over $\mathbb{F}_2$). We require that for any QPT adversary $\mathcal{A}$, if $(\mathsf{ek},\mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$ then
   $$\Pr[\mathcal{A}(\mathsf{ek}) \in W_{\mathsf{ek}}] \leq \mathrm{negl},$$
   where the probability is taken over all randomness in the experiment.

3.1 XOR Lemmas for Adaptive Hardcore Bits

The weak version (direct-product) of the adaptive hardcore bit property (property 3c) will not be sufficient for our protocol. In the following, we define a stronger version of the property that we will need in our analysis. Note that the only difference with respect to property 3c is that the adversary is required to output a single bit h, which is the XOR of the n bits required before.

Definition 6 (Adaptive Hardcore Bit). *Let $\mathcal{F}$ be a wTCF, and let W_{ek} be a (key-dependent) set of tuples defined as*

$$W_{\mathsf{ek}} = \left\{ \left(b, \mathbf{x}_b, \mathbf{d}, h\right) \middle| \begin{array}{l} b \in \{0,1\}, (\mathbf{x}_0, \mathbf{x}_1) \in R_{\mathsf{ek}}, \mathbf{d} \in \mathbb{Y}_{0,\mathbf{x}_0} \cap \mathbb{Y}_{1,\mathbf{x}_1}, \\ h = \bigoplus_{i=1}^{n} \langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \rangle \end{array} \right\},$$

where $\mathbf{d}_i$ and $\mathsf{B}_i(\cdot)$ denote the ith ℓ-bit chunk of $\mathbf{d}$ and $\mathsf{B}(\cdot)$, respectively. We require that for any QPT adversary $\mathcal{A}$, if $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$ then

$$\Pr[\mathcal{A}(\mathsf{ek}) \in W_{\mathsf{ek}}] \leq 1/2 + \mathrm{negl},$$

where the probability is taken over all randomness in the experiment.

We define the following property for a wTCF family, which requires that any key/input/output can be viewed as n independent instances. Our construction of wTCF will satisfy this property.

Definition 7. *Let $\mathcal{F}$ be a wTCF family of functions with domain X^n and range $Y = \bar{Y}^n$. Let* Gen, Eval, *and* Invert *be the associated algorithms. We say that $\mathcal{F}$ is a wTCF family with independent evaluations (wTCF-IE) if there exists algorithms* $\overline{\mathsf{Gen}}$, $\overline{\mathsf{Eval}}$, *and* $\overline{\mathsf{Invert}}$ *such that*

- Gen *is identically distributed to the concatenation of n independent runs of* $\overline{\mathsf{Gen}}$.
- *For each $(\mathsf{ek}, \mathsf{td}) = \{(\mathsf{ek}_i, \mathsf{td}_i)\}_{i\in[n]}$ in the support of $\mathsf{Gen} \equiv (\overline{\mathsf{Gen}})^n$, the output of any function $f_{\mathsf{ek},b} \in \mathcal{F}$ on any $\mathbf{x} \in X^n$ is identical to the concatenation of $\overline{\mathsf{Eval}}_{\mathsf{ek}_i,b}$ on x_i for $i \in [n]$.*
- *For each $(\mathsf{ek}, \mathsf{td}) = \{(\mathsf{ek}_i, \mathsf{td}_i)\}_{i\in[n]}$ in the support of $\mathsf{Gen} \equiv (\overline{\mathsf{Gen}})^n$, the output of $\mathsf{Invert}_{\mathsf{td},b}$ on any $\mathbf{y} \in \bar{Y}^n$ is identical to the concatenation of $\overline{\mathsf{Invert}}_{\mathsf{td}_i,b}$ on y_i for $i \in [n]$.*

Next we state our conjecture, namely that any wTCF-IE that satisfies direct-product adaptive hardcore bit property (3c), also satisfies the adaptive hardcore bit property.

Conjecture 2. If $\mathcal{F}$ is a wTCF-IE family that satisfies the direct-product adaptive hardcore bit property 3c, then $\mathcal{F}$ satisfies the property 6.

Remark 3. Note that for our construction, the conjecture above is implied by the (informal) conjecture 1 via a transformation (from claw-based inner product to shift-based equation) that we will see later. We omit the formal details as it is going to be similar to the proof of Lemma 7.

Random Subset Adaptive Hardcore Bit. To gain confidence in our conjecture, we show that a weaker variant of it is implied by property 3c. Roughly speaking, this notion says that it is hard to predict the XOR of a random subset of the adaptive n hardcore bits. However, note that the predictor is not given the subset ahead of time.

Definition 8 (Random Subset Adaptive Hardcore Bit). *Let $\mathcal{F}$ be a wTCF. For any QPT adversary $\mathcal{A}$, the success probability in the following experiment is negligibly close to $1/2$.*

- *The challenger samples $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$ and sends ek to $\mathcal{A}$.*
- *$\mathcal{A}$ sends a tuple $(b, \mathbf{x}_b, \mathbf{d})$.*
- *The challenger samples a subset $\mathbf{r} \leftarrow \{0,1\}^n$ and sends $\mathbf{r}$ to $\mathcal{A}$.*
- *$\mathcal{A}$ returns a bit $h \in \{0,1\}$ and succeeds if the following conditions are satisfied:*
  - $(\mathbf{x}_0, \mathbf{x}_1) \in R_{\mathsf{ek}}$
  - $\mathbf{d} \in \mathbb{Y}_{0,\mathbf{x}_0} \cap \mathbb{Y}_{1,\mathbf{x}_1}$
  - $h = \bigoplus_{i=1}^{n} r_i \cdot \langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \rangle$

  where $\mathbf{d}_i$ and $\mathsf{B}_i(\cdot)$ denote the ith ℓ-bit chunk of $\mathbf{d}$ and $\mathsf{B}(\cdot)$, respectively.

Next we show that this new variant is directly implied by definition 5. This is an almost immediate application of a theorem from [AC02].

Lemma 1. *Let $\mathcal{F}$ be a wTCF, then $\mathcal{F}$ satisfies definition 8.*

Proof. The proof consists of a reduction to the direct-product adaptive hardcore bit property of the wTCF (property 3c). Let $\mathcal{A}$ be a QPT algorithm that succeeds in the above game with probability greater than $1/2 + \varepsilon$, for some inverse-polynomial ε. Let $|\psi\rangle$ denote the internal state of the adversary after the second step of the protocol, and in particular after the tuple $(b, \mathbf{x}_b, \mathbf{d})$ has been sent to the challenger. Let G_{ek} be a set defined as follows:

$$G_{\mathsf{ek}} = \left\{ (b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) : \Pr\left[\mathcal{A}(\mathbf{r}; |\psi\rangle) = \bigoplus_{i=1}^{n} r_i \cdot \langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \rangle \right] \geq 1/2 + \varepsilon/2 \right\}$$

where the probability is taken over the random choice of $\mathbf{r}$ and over the internal coins of $\mathcal{A}$. We use the abbreviation $\mathcal{A}(\mathbf{r}; |\psi\rangle)$ to denote the output of the adversary $\mathcal{A}$ run on state $|\psi\rangle$ and on input $\mathbf{r}$. Observe that the above set is well-defined, since $\mathbf{x}_b$ uniquely determines the claw $(\mathbf{x}_0, \mathbf{x}_1)$, provided that $(\mathbf{x}_0, \mathbf{x}_1) \in R_{\mathsf{ek}}$.

We argue that $\Pr[(b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \in G_{\mathsf{ek}}] \geq \varepsilon/2$, where the probability is over the random choice of ek and the random coins of $\mathcal{A}$. For notational convenience, we relabel $h_i = \mathbf{d}_i \cdot \left[\mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \right]$.

Assume towards contradiction that $\Pr[(b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \in G_{\mathsf{ek}}] < \varepsilon/2$. We can then rewrite:

$$\begin{aligned}
\Pr\left[\mathcal{A} \text{ succeeds}\right] &= \Pr\left[\mathcal{A}(\mathbf{r}; |\psi\rangle) = \bigoplus_{i=1}^{n} r_i \cdot h_i\right] \\
&= \Pr\left[\mathcal{A}(\mathbf{r}; |\psi\rangle) = \bigoplus_{i=1}^{n} r_i \cdot h_i \middle| (b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \in G_{\mathsf{ek}}\right] \Pr\left[(b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \in G_{\mathsf{ek}}\right] \\
&+ \Pr\left[\mathcal{A}(\mathbf{r}; |\psi\rangle) = \bigoplus_{i=1}^{n} r_i \cdot h_i \middle| (b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \notin G_{\mathsf{ek}}\right] \Pr\left[(b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \notin G_{\mathsf{ek}}\right] \\
&< \varepsilon/2 + (1/2 + \varepsilon/2) \\
&= 1/2 + \varepsilon
\end{aligned}$$

which contradicts our initial hypothesis. Conditioned on $(b, \mathbf{x}_b, \mathbf{d}, |\psi\rangle) \in G_{\mathsf{ek}}$, we then consider the algorithm $\mathcal{A}(\cdot; |\psi\rangle)$. Such an algorithm runs in polynomial time and, on input $\mathbf{r}$, it returns

$$h = \bigoplus_{i=1}^{n} r_i \cdot h_i = \bigoplus_{i=1}^{n} r_i \cdot \langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \rangle$$

with probability at least $\varepsilon/2$ (over the random choice of $\mathbf{r}$ and the internal coins of $\mathcal{A}$). By the Adcock-Cleve theorem [AC02], it follows that there exists an efficient algorithm that, with a *single query* to $\mathcal{A}(\cdot; |\psi\rangle)$, returns $(h_1, \ldots, h_n)$ with inverse polynomial probability. This violates the direct-product adaptive hardcore bit property of $\mathcal{F}$.

4 wTCF from Extended LHS Assumption

Here we show how to construct a wTCF family from extended LHS assumption (Definition 4) over a group action $(\mathbb{G}, \mathbb{X}, \star)$.

Construction. Let n be the secret dimension of underlying extended LHS assumption, and let $B > 2n^3$ be an integer. We define a wTCF family as follows. Let $X = [B]^n$, and $Y = (\mathbb{X}^{2n})^n$. Note that $X^n = ([B]^n)^n$ and Y will be the input and output space of our wTCF family, respectively. To generate a key-trapdoor pair, for each $i \in [n]$ and $\beta \in \{0, 1\}$ sample

$$\mathbf{v}_i \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n \times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n,$$

and set

$$\begin{aligned}
\mathbf{x}_i^{(1)} &:= \left[\mathbf{M}_i^{(0)}(\mathbf{1} - \mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)}, \\
\mathbf{y}_i^{(1)} &:= \left[\mathbf{M}_i^{(0)}(\mathbf{1} - \mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i + \mathbf{m}_i^{(0)} \odot (\mathbf{1} - \mathbf{v}_i) + \mathbf{m}_i^{(1)} \odot \mathbf{v}_i\right] \star \mathbf{y}_i^{(0)},
\end{aligned}$$

where $\odot$ denotes component-wise product. Output $(\mathsf{ek}, \mathsf{td})$ where

$$\mathsf{td} = \left(\mathbf{v}_i, \mathbf{t}_i\right)_{i\in[n]}, \quad \mathsf{ek} = \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}.$$

To evaluate the function $f_{\mathsf{ek},b}$ on input $(\mathbf{s}_i)_{i\in[n]} \in ([B]^n)^n$, output $(\bar{\mathbf{z}}_i, \mathbf{z}_i)$ for $i \in [n]$ where

$$\bar{\mathbf{z}}_i = \left[(1-b)\cdot \mathbf{M}_i^{(0)}\mathbf{1} + \left(\mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}\right)\mathbf{s}_i\right] \star \mathbf{x}_i^{(b)},$$
$$\mathbf{z}_i = \left[(1-b)\cdot \mathbf{M}_i^{(0)}\mathbf{1} + \left(\mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}\right)\mathbf{s}_i + (1-b)\cdot \mathbf{m}_i^{(0)} + (\mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}) \odot \mathbf{s}_i\right] \star \mathbf{y}_i^{(b)}.$$

To invert the function $f_{\mathsf{ek},b}$ on some value $(\bar{\mathbf{z}}_i, \mathbf{z}_i)_{i\in[n]}$, we recover each $\mathbf{s}_i$ (for $i \in [n]$) as follows. Observe that if $f_{\mathsf{ek},b}((\mathbf{s}_i)_{i\in[n]}) = (\bar{\mathbf{z}}_i, \mathbf{z}_i)_{i\in[n]}$ then the following relation holds for any $i \in [n]$:

$$(-\mathbf{t}_i - \mathbf{m}_i^{(0)}) \star \mathbf{z}_i = \left[(\mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}) \odot (\mathbf{s}_i + b\cdot \mathbf{v}_i)\right] \star \bar{\mathbf{z}}_i.$$

Because the action is applied component-wise and each entry of $\mathbf{s}_i$ lies in $[B]$, one can recover each entry of $\mathbf{s}_i$ efficiently by a simple brute force, since both $\mathbf{v}_i$ and $\mathbf{t}_i$ are included in the trapdoor.

We have already shown the construction above satisfies the properties (1) and (2a) of a wTCF family, thus proving the following lemma.

Lemma 2. *Let $\mathcal{F}$ be the function family (with associated algorithms) as described in the construction, then $\mathcal{F}$ satisfies the properties 1 and 2a.*

Next, we show the construction above satisfies the remaining properties of a wTCF family (Definition 5) via the following lemmata.

Lemma 3. *Let $\mathcal{F}$ be the function family (with associated algorithms) as described in the construction, then $\mathcal{F}$ satisfies the property 2b.*

Proof. It is easy to see that for all but a negligible fraction of key-trapdoor pairs $(\mathsf{ek}, \mathsf{td}) \in \mathrm{supp}(\mathsf{Gen}(1^\lambda))$

- Any evaluation key ek uniquely determines $\mathbf{v}_i$ for $i \in [n]$.[1]
- $f_{\mathsf{ek},b}$ is an injective function.

For a given evaluation key ek, consider the following two subsets:

$$\mathbf{X}_0 = \left\{(\mathbf{s}_i)_{i\in[n]} \mid \forall i \in [n] : \mathbf{s}_i \in [B]^n \wedge \mathbf{s}_i - \mathbf{v}_i \in [B]^n\right\},$$
$$\mathbf{X}_1 = \left\{(\mathbf{s}_i)_{i\in[n]} \mid \forall i \in [n] : \mathbf{s}_i \in [B]^n \wedge \mathbf{s}_i + \mathbf{v}_i \in [B]^n\right\}.$$

Let $R_{\mathsf{ek}} \subseteq \mathbf{X}_0 \times \mathbf{X}_1$ be the relation defined as

$$R_{\mathsf{ek}} = \left\{\left(\left(\mathbf{s}_i^{(0)}\right)_{i\in[n]}, \left(\mathbf{s}_i^{(1)}\right)_{i\in[n]}\right) \in \mathbf{X}_0 \times \mathbf{X}_1 \,\middle|\, \forall i \in [n] : \mathbf{s}_i^{(0)} = \mathbf{s}_i^{(1)} + \mathbf{v}_i\right\}.$$

[1] Recall that $\mathbb{G}$ is a superpolynomially (and possibly exponentially) large group. For example, in case of the variant from [BKV19] the group is cyclic, and hence a randomly chosen evaluation key uniquely determines $\mathbf{v}_i$ with overwhelming probability [BM87].

One can immediately verify that R_{ek} is a perfect matching. Because $f_{\mathsf{ek},b}$ is injective, it holds that

$$\begin{aligned}\forall\Big(\big(\mathbf{s}_i^{(0)}\big)_{i\in[n]}, \big(\mathbf{s}_i^{(1)}\big)_{i\in[n]}\Big) \in \mathbf{X}_0 \times \mathbf{X}_1 :& \\ f_{\mathsf{ek},0}\Big(\big(\mathbf{s}_i^{(0)}\big)_{i\in[n]}\Big) = f_{\mathsf{ek},1}\Big(\big(\mathbf{s}_i^{(1)}\big)_{i\in[n]}\Big) \quad &\iff \\ \big(\mathbf{s}_i^{(0)}\big)_{i\in[n]} = \big(\mathbf{s}_i^{(1)} + \mathbf{v}_i\big)_{i\in[n]} \quad &\iff \\ \Big(\big(\mathbf{s}_i^{(0)}\big)_{i\in[n]}, \big(\mathbf{s}_i^{(1)}\big)_{i\in[n]}\Big) \in R_{\mathsf{ek}}.&\end{aligned}$$

Since each $\mathbf{v}_i$ is a binary vector, it follows that for any $b \in \{0,1\}$

$$\Pr_{(\mathbf{s}_i)_{i\in[n]}\leftarrow([B]^n)^n}\big[(\mathbf{s}_i)_{i\in[n]} \in \mathbf{X}_b\big] \geq 1 - n^2(B-1)^{-1} \geq 1 - n^{-1}.$$

For any $b \in \{0,1\}$, given any tuple $(\mathbf{s}_i)_{i\in[n]}$ membership in $\mathbf{X}_b$ can be checked efficiently using the trapdoor, simply by testing whether $\mathbf{s}_i - (-1)^b\mathbf{v}_i \in [B^n]$ for all $i \in [n]$.

Finally, define the set $\bar{\mathbf{X}}$ as

$$\bar{\mathbf{X}} = \big\{(\mathbf{s}_i)_{i\in[n]} \mid \forall i \in [n] : \mathbf{s}_i \in \{2,\ldots,B-1\}^n\big\}.$$

Membership in $\bar{\mathbf{X}}$ can be checked efficiently without a trapdoor. Moreover, by a simple argument, we have

$$\Pr_{(\mathbf{s}_i)_{i\in[n]}\leftarrow([B]^n)^n}\big[(\mathbf{s}_i)_{i\in[n]} \in \bar{\mathbf{X}}\big] \geq 1 - 2n^2(B-1)^{-1} \geq 1 - n^{-1},$$

and hence $\bar{\mathbf{X}}$ is a dense subset of the input space $([B]^n)^n$.

Lemma 4. *Let $\mathcal{F}$ be the function family (with associated algorithms) as described in the construction, then $\mathcal{F}$ satisfies the properties 3a and 3b.*

Proof. Consider the binary encoding function $\mathsf{B} : (([B])^n)^n \to \{0,1\}^{n\ell}$ where $\ell = n\lceil\log B\rceil$. Specifically, $\mathsf{B}((\mathbf{s}_i)_{i\in[n]})$ outputs the binary representation of $(\mathbf{s}_i)_{i\in[n]}$, where each component of $\mathbf{s}_i$ is represented using a chunk of $\lceil\log B\rceil$-bit string. It is immediate to see that B is injective and it is also efficiently invertible on its range, and hence $\mathcal{F}$ satisfies the property 3a.

To avoid abusing the notation, we also define a simple function $\bar{\mathsf{B}} : [B] \to \{0,1\}^{\lceil\log B\rceil}$, which outputs the binary representation of any $s \in [B]$. For a tuple $(b, \mathbf{s}, \mathbf{d}) \in \{0,1\} \times [B]^n \times \{0,1\}^\ell$, let $\mathsf{T}_{b,\mathbf{d}} : [B]^n \to \{0,1\}^n$ be a function that maps $\mathbf{s} = (s_1,\ldots,s_n)$ to $\mathbf{d}' = (d'_1,\ldots,d'_n)$ where

$$d'_j = \langle \mathbf{d}^{(j)}, \bar{\mathsf{B}}(s_j) \oplus \bar{\mathsf{B}}(s_j - (-1)^b)\rangle, \quad j \in [n],$$

and $\mathbf{d}^{(j)}$ denotes the jth $\lceil\log B\rceil$-bit chunk of $\mathbf{d}$. Note that the inner product is computed over $\mathbb{F}_2$, while the operation $-$ is performed over $\mathbb{Z}$. As we will see later, the motivation for defining the transformation $\mathsf{T}_{b,\mathbf{d}}$ stems from the following observation [BCM+18]

that given $(\mathbf{s}, \mathbf{d}, \langle \mathbf{d}, (\mathbf{s}+\mathbf{v})\rangle) \in [B]^n \times \{0,1\}^\ell \times \{0,1\}$ for some binary $\mathbf{v} \in \{0,1\}^n$, where the inner product is computed over $\mathbb{F}_2$ and the addition $+$ is over integers, one can use $\mathsf{T}_{b,\mathbf{d}}$ to obtain a pair of the form $(\mathbf{d}', \langle \mathbf{d}', \mathbf{v}\rangle) \in \{0,1\}^n \times \{0,1\}$. This transformation will be useful in proving the weak adaptive hardcore bit property 3c.

For any $\mathbf{s} \in [B]^n$, since $\bar{\mathsf{B}}$ is an injective function it follows that the term $\bar{\mathsf{B}}(s_j) \oplus \bar{\mathsf{B}}(s_j - (-1)^b)$ is non-zero for any $j \in [n]$. Therefore, if $\mathbf{d}^{(j)} \leftarrow \{0,1\}^{\lceil \log B \rceil}$ then d'_j will be 0 with probability 1/2. It follows that for any $\mathbf{s} \in [B]^n$ and any $b \in \{0,1\}$, if $\mathbf{d} \leftarrow \{0,1\}^\ell$ then

$$\Pr[\mathsf{T}_{b,\mathbf{d}}(\mathbf{s}) = 0^n] \leq \mathrm{negl}\,.$$

For any $b \in \{0,1\}$ and any $(\mathbf{s}_i)_{i\in[n]} \in ([B]^n)^n$, consider the following set

$$\mathbb{Y}_{b,(\mathbf{s}_i)_{i\in[n]}} = \Big\{ (\mathbf{d}_i)_{i\in[n]} \in \{0,1\}^{n\ell} \Big| \; \forall i \in [n] : \mathsf{T}_{b,\mathbf{s}_i}(\mathbf{d}_i) \neq 0^n \Big\}.$$

By a simple union bound it follows that for any $b \in \{0,1\}$ and $(\mathbf{s}_i)_{i\in[n]} \in ([B]^n)^n$ we have

$$\Pr_{(\mathbf{d}_i)_{i\in[n]} \leftarrow \{0,1\}^{n\ell}} \left[(\mathbf{d}_i)_{i\in[n]} \in \mathbb{Y}_{b,(\mathbf{s}_i)_{i\in[n]}} \right] \leq \mathrm{negl},$$

where we used the fact that for each $i \in [n]$ it holds that $\Pr_{\mathbf{d}_i}[\mathsf{T}_{b,\mathbf{d}_i}(\mathbf{s}_i) = 0^n] \leq \mathrm{negl}$. Clearly, $\mathsf{T}_{b,\mathbf{d}}$ is efficiently computable, and hence membership in $\mathbb{Y}_{b,(\mathbf{s}_i)_{i\in[n]}}$ is efficiently checkable given b and $(\mathbf{s}_i)_{i\in[n]}$, establishing the property 3b.

Lemma 5. *Let $\mathcal{F}$ be the function family (with associated algorithms) as described in the construction, then $\mathcal{F}$ satisfies the property 3c based on the extended LHS assumption.*

Proof. The lemma follows from putting together Lemma 6 (proving hardness of an alternative formulation of the extended LHS assumption), Lemma 7 (which shows a transformation relating claw-based equations to linear equations in $\mathbf{v}_i$), and Lemma 8 (showing hardness of predicting concatenation of any non-trivial parity of $\mathbf{v}_i$ for $i \in [n]$ based on the extended LHS assumption), all of which will be proved subsequently.

Theorem 1. *Let $\mathcal{F}$ be the function family (with associated algorithms) as described in the construction, then $\mathcal{F}$ is a wTCF-IE family based on the extended LHS assumption.*

Proof. We have already established that $\mathcal{F}$ is a wTCF family by putting together Lemma 2, Lemma 3, Lemma 4, and Lemma 5. It follows by inspection that $\mathcal{F}$ also satisfies the independent evaluation property 7, and hence $\mathcal{F}$ is a wTCF-IE family.

The following lemma establishes the hardness of a different formulation of the extended LHS assumption.

Lemma 6. *If H_0 and H_1 be two distributions defined as follows then $H_0 \overset{c}{\approx} H_1$ based on the extended LHS assumption.*

$$\mathbf{w} \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n,$$
$$\mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n, \quad \mathbf{u}_i^{(\beta)} \leftarrow \mathbb{X}^n, \quad \mathbf{u'}_i^{(\beta)} \leftarrow \mathbb{X}^n,$$

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{w}) + \mathbf{M}_i^{(1)}\mathbf{w}\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{w}) + \mathbf{M}_i^{(1)}\mathbf{w} + \mathbf{m}_i^{(0)} \odot (\mathbf{1}-\mathbf{w}) + \mathbf{m}_i^{(1)} \odot \mathbf{w}\right] \star \mathbf{y}_i^{(0)},$$

$$H_0 := \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}, \quad H_1 := \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{u}_i^{(\beta)}, \mathbf{u'}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}},$$

Proof. Given a challenge of the form

$$H' = \left(\mathbf{M}_i, \mathbf{m}_i, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}},$$

the reduction samples two matrices $\mathbf{M}_i^{(0)}$ and $\mathbf{M}_i^{(1)}$ and two vectors $\mathbf{m}_i^{(0)}$ and $\mathbf{m}_i^{(1)}$ uniformly *conditioned* on

$$\mathbf{M}_i = \mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}, \quad \mathbf{m}_i = \mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}.$$

It then sets

$$\bar{\mathbf{x}}_i^{(0)} := \mathbf{x}_i^{(0)}, \qquad \bar{\mathbf{y}}_i^{(0)} := \mathbf{y}_i^{(0)},$$
$$\bar{\mathbf{x}}_i^{(1)} := \mathbf{M}_i^{(0)}\mathbf{1} \star \mathbf{x}_i^{(0)}, \qquad \mathbf{y}_i^{(1)} := \left[\mathbf{M}_i^{(0)}\mathbf{1} + \mathbf{m}_i^{(0)} \odot \mathbf{1}\right] \star \mathbf{y}_i^{(0)},$$

and outputs the following tuple

$$\left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \bar{\mathbf{x}}_i^{(\beta)}, \bar{\mathbf{y}}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}.$$

Observe that in the tuple above $\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}$ are distributed uniformly for $i \in [n]$ and $\beta \in \{0,1\}$. If H' corresponds to extended LHS samples, a routine calculation shows that the tuple above is distributed as H_0. On the other hand, if H' corresponds to truly random samples then the tuple above would be distributed as H_1. Therefore, based on the extended LHS assumption it follows that H_0 is indistinguishable from H_1.

Lemma 7. *Let* $\mathsf{ek} = \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}$ *be a tuple distributed as in the construction, i.e.,*

$$\mathbf{v}_i \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n,$$

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i + \mathbf{m}_i^{(0)} \odot (\mathbf{1}-\mathbf{v}_i) + \mathbf{m}_i^{(1)} \odot \mathbf{v}_i\right] \star \mathbf{y}_i^{(0)},$$

and let W_{ek} *be the set defined in the property 3c with respect to the construction of wTCF family, i.e.,*

$$W_{\mathsf{ek}} = \Big\{ \Big(b, \left(\mathbf{s}_i^{(b)}\right)_{i\in[n]}, \mathbf{d}, \left(\langle \mathbf{d}_i, \mathsf{B}_i\left(\left(\mathbf{s}_i^{(0)}\right)_{i\in[n]}\right) \oplus \mathsf{B}_i\left(\left(\mathbf{s}_i^{(1)}\right)_{i\in[n]}\right)\rangle\right)_{i\in[n]} \;\Big|$$
$$b \in \{0,1\}, \left(\left(\mathbf{s}_i^{(0)}\right)_{i\in[n]}\right), \left(\mathbf{s}_i^{(1)}\right)_{i\in[n]}\right)\Big) \in R_{\mathsf{ek}}, \mathbf{d} \in \mathbb{Y}_{0,(\mathbf{s}_i^{(0)})_{i\in[n]}} \cap \mathbb{Y}_{1,(\mathbf{s}_i^{(1)})_{i\in[n]}} \Big\},$$

where B, R_{ek}, *and* $\mathbb{Y}_{b,(\mathbf{s}_i)_{i\in[n]}}$ *are defined in the proof of Lemma 4. If there is an attacker* $\mathcal{A}$ *such that*

$$\Pr[\mathcal{A}(\mathsf{ek}) \in W_{\mathsf{ek}}] = \varepsilon,$$

then there is an attacker $\mathcal{A}'$ *such that*

$$\Pr\left[\mathcal{A}'(\mathsf{ek}) \to \left(\mathbf{d}'_i \neq 0^n, \langle \mathbf{d}'_i, \mathbf{v}_i\rangle\right)_{i\in[n]}\right] \geq \varepsilon.$$

Proof. Let the following tuple

$$\gamma := \left(b, \left(\mathbf{s}_i^{(b)}\right)_{i\in[n]}, \mathbf{d}, \left(c_i\right)_{i\in[n]}\right),$$

be the output $\mathcal{A}$ on ek. We are going to argue that if $\gamma \in W_{\mathsf{ek}}$ and $\mathbf{d}'_i$ is computed as $\mathbf{d}'_i = \mathsf{T}_{b,\mathbf{s}_i^{(b)}}(\mathbf{d}_i)$ for $i \in [n]$, then $c_i = \langle \mathbf{d}'_i, \mathbf{v}_i\rangle$ for all $i \in [n]$, where T is the transformation defined in the proof of Lemma 4. Observe that since

$$\mathbf{d} \in \mathbb{Y}_{0,(\mathbf{s}_i^{(0)})_{i\in[n]}} \cap \mathbb{Y}_{1,(\mathbf{s}_i^{(1)})_{i\in[n]}} \subseteq \mathbb{Y}_{b,(\mathbf{s}_i^{(b)})_{i\in[n]}},$$

it follows from the definition of these sets (in the proof of Lemma 4) that for each $i \in [n]$ we have $\mathbf{d}'_i \neq 0^n$. Furthermore, relying again on the proof of Lemma 4 we have

$$\left(\left(\mathbf{s}_i^{(0)}\right)_{i\in[n]}, \left(\mathbf{s}_i^{(1)}\right)_{i\in[n]}\right) \in R_{\mathsf{ek}} \implies \left(\mathbf{s}_i^{(0)}\right)_{i\in[n]} = \left(\mathbf{s}_i^{(1)} + \mathbf{v}_i\right)_{i\in[n]} \implies$$
$$\mathsf{B}\left(\left(\mathbf{s}_i^{(0)}\right)_{i\in[n]}\right) \oplus \mathsf{B}\left(\left(\mathbf{s}_i^{(1)}\right)_{i\in[n]}\right) = \mathsf{B}\left(\left(\mathbf{s}_i^{(b)}\right)_{i\in[n]}\right) \oplus \mathsf{B}\left(\left(\mathbf{s}_i^{(1-b)} - (-1)^b\mathbf{v}_i\right)_{i\in[n]}\right).$$

Let $d'_{i,j}$, $s_{i,j}^{(b)}$, and $v_{i,j}$ be the jth component of $\mathbf{d}'_i$, $\mathbf{s}_i^{(b)}$, and $\mathbf{v}_i$, respectively. Let $\mathbf{d}_{i,j} \in \{0,1\}^{\lceil \log B\rceil}$ be the jth $\lceil \log B\rceil$-bit chunk of $\mathbf{d}_i$. By definition of T and $\bar{\mathsf{B}}$ from the proof of Lemma 4, it follows that for any $i \in [n]$ we have

$$\begin{aligned} c_i &= \sum_{j=1}^{n} \langle \mathbf{d}_{i,j}, \left(\bar{\mathsf{B}}(s_{i,j}^{(b)}) \oplus \bar{\mathsf{B}}(s_{i,j}^{(b)} - (-1)^b v_{i,j})\right)\rangle \\ &= \sum_{j=1}^{n} v_{i,j} \langle \mathbf{d}_{i,j}, \left(\bar{\mathsf{B}}(s_{i,j}^{(b)}) \oplus \bar{\mathsf{B}}(s_{i,j}^{(b)} - (-1)^b)\right)\rangle \\ &= \sum_{j=1}^{n} v_{i,j} d'_j = \langle \mathbf{d}'_i, \mathbf{v}_i\rangle, \end{aligned}$$

where the second line follows from the fact that $v_{i,j} \in \{0,1\}$ and the last line follows from the definition of T. Note that any computation inside $\bar{\mathsf{B}}$ is done over $\mathbb{Z}$, while any other computation (including the overall summation) is performed over $\mathbb{F}_2$.

Viewing any evaluation key ek as a (one-way) function of $(\mathbf{v}_i)_{i\in[n]}$ in the construction, the following lemma establishes that any QPT adversary cannot predict a string obtained by concatenating *any* non-trivial parity of $\mathbf{v}_i$ for $i \in [n]$.

Lemma 8. *If* $\mathsf{ek} = \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}$ *be a tuple distributed as in the construction, i.e.,*

$$\mathbf{v}_i \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n,$$

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{v}_i) + \mathbf{M}_i^{(1)}\mathbf{v}_i + \mathbf{m}_i^{(0)} \odot (\mathbf{1}-\mathbf{v}_i) + \mathbf{m}_i^{(1)} \odot \mathbf{v}_i\right] \star \mathbf{y}_i^{(0)},$$

then for any QPT adversary $\mathcal{A}$ *we have*

$$\Pr\left[\mathcal{A}(\mathsf{ek}) \to \left(\mathbf{d}_i' \neq 0^n, \langle \mathbf{d}_i', \mathbf{v}_i\rangle\right)_{i\in[n]}\right] \leq \mathrm{negl},$$

where the probability is taken over randomness of ek *and* $\mathcal{A}$*, and the inner product is computed over* $\mathbb{F}_2$*.*

Proof. Consider the following two hybrids H_0 and H_1 defined as

$$\mathbf{w} \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n,$$

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{w}) + \mathbf{M}_i^{(1)}\mathbf{w}\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \left[\mathbf{M}_i^{(0)}(\mathbf{1}-\mathbf{w}) + \mathbf{M}_i^{(1)}\mathbf{w} + \mathbf{m}_i^{(0)} \odot (\mathbf{1}-\mathbf{w}) + \mathbf{m}_i^{(1)} \odot \mathbf{w}\right] \star \mathbf{y}_i^{(0)},$$

$$H_0 := \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}, \quad H_1 := \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{u}_i^{(\beta)}, \mathbf{u'}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}},$$

where $\mathbf{u}_i^{(\beta)} \leftarrow \mathbb{X}^n$ and $\mathbf{u'}_i^{(\beta)} \leftarrow \mathbb{X}^n$ for $i \in [n]$ and $\beta \in \{0,1\}$. Note that H_0 does *not* correspond to the distribution of a "real" evaluation key, as H_0 incorporates a *single* vector $\mathbf{w} \in \{0,1\}^n$ across different samples. We show that given any adversary with a non-negligible advantage in outputting concatenation of adaptive hardcore bits, one can construct another adversary that can distinguish between H_0 and H_1 with a non-negligible advantage. By Lemma 6, we know that H_0 is computationally distinguishable from H_1 and hence the statement of the lemma follows.

For any vector $\mathbf{r} \in \{0,1\}^n$, let $\pi_{\mathbf{r}}$ be a simple mapping that takes two n by n matrices $\mathbf{M}^{(0)}$ and $\mathbf{M}^{(1)}$, and for each $i \in [n]$ it *swaps* the ith column of $\mathbf{M}^{(0)}$ and $\mathbf{M}^{(1)}$ if $r_i = 1$. As two simple examples, we have

$$\pi_{0^n}\left(\mathbf{M}^{(0)}, \mathbf{M}^{(1)}\right) = \left(\mathbf{M}^{(0)}, \mathbf{M}^{(1)}\right), \quad \pi_{1^n}\left(\mathbf{M}^{(0)}, \mathbf{M}^{(1)}\right) = \left(\mathbf{M}^{(1)}, \mathbf{M}^{(0)}\right).$$

As a simple special case, we also use the notation $\pi_{\mathbf{r}}(\mathbf{m}^{(0)}, \mathbf{m}^{(1)})$ to denote swapping components of two vectors $\mathbf{m}^{(0)}$ and $\mathbf{m}^{(1)}$ with respect to $\mathbf{r}$. Let the following

$$\left(\mathbf{M}_i^{(0)}, \mathbf{M}_i^{(1)}, \mathbf{m}_i^{(0)}, \mathbf{m}_i^{(1)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}$$

be a tuple that is distributed as H_0 (with a slight reformatting). For any n binary vectors $\mathbf{r}_i \in \{0,1\}^n$, set

$$\left(\mathbf{M'}_i^{(0)}, \mathbf{M'}_i^{(1)}\right) := \pi_{\mathbf{r}_i}\left(\mathbf{M}_i^{(0)}, \mathbf{M}_i^{(1)}\right), \qquad \left(\mathbf{m'}_i^{(0)}, \mathbf{m'}_i^{(1)}\right) := \pi_{\mathbf{r}_i}(\mathbf{m}_i^{(0)}, \mathbf{m}_i^{(1)}),$$

and observe that the tuple

$$H'_0 := \left(\mathbf{M'}_i^{(0)}, \mathbf{M'}_i^{(1)}, \mathbf{m'}_i^{(0)}, \mathbf{m'}_i^{(1)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}$$

is distributed as follows:

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M'}_i^{(0)}(\mathbf{1} - (\mathbf{w} \oplus \mathbf{r}_i)) + \mathbf{M'}_i^{(1)}(\mathbf{w} \oplus \mathbf{r}_i)\right] \star \mathbf{x}_i^{(0)},$$
$$\mathbf{y}_i^{(1)} := \left[\mathbf{M'}_i^{(0)}(\mathbf{1} - (\mathbf{w} \oplus \mathbf{r}_i)) + \mathbf{M'}_i^{(1)}(\mathbf{w} \oplus \mathbf{r}_i) + \mathbf{m'}_i^{(0)} \odot (\mathbf{1} - (\mathbf{w} \oplus \mathbf{r}_i)) + \mathbf{m'}_i^{(1)} \odot (\mathbf{w} \oplus \mathbf{r}_i)\right] \star \mathbf{y}_i^{(0)}.$$

Now if we sample each $\mathbf{r}_i$ randomly, it is not hard to see that H'_0 is *statistically* indistinguishable from an honestly generated evaluation ek as defined in the lemma. Thus, there is an efficient randomized procedure $\mathcal{P}$ that maps an instance of H_0 to an honestly generated ek. Furthermore, applying the same procedure $\mathcal{P}$ would still map an instance of H_1 to an instance of H_1.

Let H_b (for some challenge $b \in \{0,1\}$) be a challenge tuple of the form

$$\left(\mathbf{M}_i^{(0)}, \mathbf{M}_i^{(1)}, \mathbf{m}_i^{(0)}, \mathbf{m}_i^{(1)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}},$$

and let $\mathcal{A}$ be an attacker that outputs concatenation of adaptive hardcore bits. We construct an adversary $\mathcal{A}'$ that distinguishes H_0 and H_1. First, $\mathcal{A}'$ samples n random vector $\mathbf{r}_i \leftarrow \{0,1\}^n$ and sets

$$\left(\mathbf{M'}_i^{(0)}, \mathbf{M'}_i^{(1)}\right) := \pi_{\mathbf{r}_i}\left(\mathbf{M}_i^{(0)}, \mathbf{M}_i^{(1)}\right), \qquad (\mathbf{m'}_i^{(0)}, \mathbf{m'}_i^{(1)}) := \pi_{\mathbf{r}_i}(\mathbf{m}_i^{(0)}, \mathbf{m}_i^{(1)}).$$

It then runs $\mathcal{A}$ on $\bar{\mathsf{ek}}$ where

$$\bar{\mathsf{ek}} = \left(\mathbf{M'}_i^{(0)}, \mathbf{M'}_i^{(1)}, \mathbf{m'}_i^{(0)}, \mathbf{m'}_i^{(1)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[n],\beta\in\{0,1\}}.$$

Let $(\mathbf{d}'_i, c_i)_{i\in[n]}$ be the output of $\mathcal{A}(\bar{\mathsf{ek}})$. In the next step $\mathcal{A}'$ proceeds as follows:

$\mathcal{A}'$ computes $c'_i = \langle \mathbf{d}'_i, \mathbf{r}_i \rangle \oplus c_i$ for $i \in [n]$. Let $\mathbf{D'} \in \{0,1\}^{n\times n}$ be a matrix whose rows are $\mathbf{d}'_i$.

- Case 1: If $(\mathbf{d}'_i)_{i\in[n]}$ are linearly independent vectors, $\mathcal{A}'$ computes $\mathbf{w'} = \mathbf{D'}^{-1}\mathbf{c'}$, where operations are performed over $\mathbb{F}_2$. If the following holds, $\mathcal{A}'$ outputs 0. Otherwise it outputs a random bit b'.

$$\mathbf{x}_1^{(1)} = \left[\mathbf{M}_1^{(0)}(\mathbf{1} - \mathbf{w'}) + \mathbf{M}_1^{(1)}\mathbf{w'}\right] \star \mathbf{x}_1^{(0)}.$$

- Case 2: There is a minimal index $n' > 1$ and $n' - 1$ bits $(\alpha_1, \ldots, \alpha_{n'-1})$ such that $\mathbf{d}'_{n'} = \sum_{i=1}^{n'-1} \alpha_i \mathbf{d}'_i$. If the following holds, $\mathcal{A}'$ outputs 0. Otherwise, it outputs a random bit b'.

$$c'_{n'} = \sum_{i=1}^{n'-1} \alpha_i c'_i.$$

We now analyze the advantage of $\mathcal{A}'$ in distinguishing H_0 and H_1.

– H_b is distributed as H_0: Since $\mathcal{A}'$ maps an instance of H_0 to a tuple that is statistically indistinguishable from an honestly generated evaluation key, it follows that if ε be the advantage of $\mathcal{A}$, then

$$\Pr\left[\mathcal{A}(\bar{\mathsf{ek}}) \to \left(\mathbf{d}'_i \neq 0^n, \langle \mathbf{d}'_i, \mathbf{w} \oplus \mathbf{r}_i \rangle\right)_{i\in[n]}\right] = \varepsilon,$$

and hence with probability ε we have

$$c'_i = \langle \mathbf{d}'_i, \mathbf{r}_i \rangle \oplus c_i = \langle \mathbf{d}'_i, \mathbf{r}_i \rangle \oplus \langle \mathbf{d}'_i, \mathbf{w} \oplus \mathbf{r}_i \rangle = \langle \mathbf{d}'_i, \mathbf{w} \rangle, \quad \forall i \in [n].$$

Furthermore, it is easy to see that conditioned on the event that $\mathcal{A}$ succeeds, $\mathcal{A}'$ outputs 0. This follows immediately by observing that in case 1, $\mathcal{A}'$ recovers $\mathbf{w}$, and in case 2

$$c'_{n'} = \langle \mathbf{d}'_{n'}, \mathbf{w} \rangle = \langle \sum_{i=1}^{n'-1} \alpha_i \mathbf{d}'_i, \mathbf{w} \rangle = \sum_{i=1}^{n'-1} \alpha_i \langle \mathbf{d}'_i, \mathbf{w} \rangle = \sum_{i=1}^{n'-1} \alpha_i c'_i.$$

Therefore, it holds that

$$\Pr[\mathcal{A}'(H_0) = 0] \geq \varepsilon + (1-\varepsilon) \cdot \Pr[b' = 0] = (1+\varepsilon)/2.$$

– H_b is distributed as H_1: Although $\mathcal{A}'$ maps a truly random instance (i.e., H_1) to a truly random instance, we can still argue that $\mathcal{A}'$ outputs 0 with probability negligibly close to 1/2. First, observe that the vectors $(\mathbf{r}_i)_{i\in[n]}$ are information-theoretically hidden from the view of $\mathcal{A}$. Thus, conditioned on the event that case 2 happens we have

$$\Pr\left[c'_{n'} = \sum_{i=1}^{n'-1} \alpha_i c'_i\right] = \Pr\Big[\underbrace{\langle \mathbf{d}'_{n'}, \mathbf{r}_{n'} \rangle \oplus c_{n'}}_{\sigma_L} = \underbrace{\sum_{i=1}^{n'-1} \alpha_i \cdot \left(\langle \mathbf{d}'_i, \mathbf{r}_i \rangle \oplus c_i\right)}_{\sigma_R}\Big].$$

Because $\mathbf{d}'_i \neq 0^n$ (for all $i \in [n]$) and there exists at least one index i^* such that $\alpha_{i^*} \neq 0$, it follows that the left-hand side (σ_L) and the right-hand side (σ_R) are distributed independently from each other, and hence we have

$$\Pr\left[c'_{n'} = \sum_{i=1}^{n'-1} \alpha_i c'_i\right] = 1/2.$$

A similar argument implies that conditioned on the event that case 1 happens, $\mathcal{A}'$ outputs 0 with probability $1/2 + \mathrm{negl}$. Therefore, it holds that

$$\Pr[\mathcal{A}'(H_0) = 0] \leq 1/2 + \mathrm{negl},$$

and hence the advantage of $\mathcal{A}'$ in distinguishing H_0 and H_1 is at least $\varepsilon/2 - \mathrm{negl}$, as required.

5 Computational Test of Qubit

We show that our wTCF can be used to devise a computational test that the prover has a qubit. The protocol closely follows the outline of [Vid20], with a few syntactical modifications, due to the usage of wTCFs.

5.1 Definition

We start by recalling the definition of a qubit. We denote $\{A, B\} \equiv AB + BA$ as the *anti-commuter* of two operators A and B, and we say A *anti-commutes* B if $\{A, B\} = 0$.

Definition 9 (Qubit). *A* qubit *is a triple* $(|\psi\rangle, X, Z)$ *such that* $|\psi\rangle$ *is a unit vector on* $\mathcal{H}$ *and* X, Z *are binary observables on* $\mathcal{H}$*, such that*

$$\{X, Z\} |\psi\rangle = 0.$$

As usual in the computational settings, we will be interested in a slightly weaker guarantee, where the above quantity is bounded by a negligible function negl, in which case we say that the tuple $(|\psi\rangle, X, Z)$ is computationally close to a qubit. The following lemma justifies the definition of a qubit, and its proof can be found in [Vid20].

Lemma 9 ([Vid20]). *Let* $(|\psi\rangle, X, Z)$ *be a qubit on* $\mathcal{H}$*. Then there exists a Hilbert space* $\mathcal{H}'$ *and an isometry* $V : \mathcal{H} \to \mathbb{C}^2 \otimes \mathcal{H}'$ *such that:*

$$VX |\psi\rangle = (\sigma_X \otimes \mathsf{Id})V |\psi\rangle \qquad \textit{and} \qquad VZ = (\sigma_Z \otimes \mathsf{Id})V |\psi\rangle$$

where

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \textit{and} \qquad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

are the Pauli observables.

5.2 Protocol

Let $\mathcal{F}$ be a wTCF function family. The protocol for a computational test of a qubit is described below.

1. The verifier samples $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^\lambda)$ and sends ek to the prover.
2. The prover prepares the state

$$\frac{1}{\sqrt{2 \cdot |X^n|}} \sum_{b \in \{0,1\}} \sum_{\mathbf{x}_b \in X^n} |b\rangle |\mathbf{x}_b\rangle |f_{\mathsf{ek},b}(\mathbf{x}_b)\rangle$$

which is efficiently computable since $f_{\mathsf{ek},b}$ is efficiently computable. Then it uncomputes the first register and traces it out to obtain

$$\frac{1}{\sqrt{2 \cdot |X^n|}} \sum_{b \in \{0,1\}} \sum_{\mathbf{x}_b \in X^n} |\mathbf{x}_b\rangle |f_{\mathsf{ek},b}(\mathbf{x}_b)\rangle .$$

Note that this mapping is efficiently computable since, given $\mathbf{x}_b$ and $f_{\mathsf{ek},b}(\mathbf{x}_b)$, the bit b is efficiently computable. The prover then measures the last register in the computational basis to obtain some $y \in Y$. The prover returns y to the verifier.

3. The verifier computes

$$\mathsf{Invert}(\mathsf{td}, 0, y) = \mathbf{x}_0 \text{ and } \mathsf{Invert}(\mathsf{td}, 1, y) = \mathbf{x}_1$$

and aborts if $\mathbf{x}_0 \notin \bar{\mathbf{X}}$ and $\mathbf{x}_1 \notin \bar{\mathbf{X}}$. The verifier then selects a uniformly random challenge $c \leftarrow \{0,1\}$ and sends c to the prover.

4. (a) (Preimage test) If $c = 0$, the prover measures the first register in the computational basis to obtain an $\mathbf{x}$, which is sent to the verifier. The verifier accepts if there exists a $b \in \{0,1\}$ such that $f_{\mathsf{ek},b}(\mathbf{x}) = y$.
   (b) (Equation test) If $c = 1$, the prover measures the first register in the Hadamard basis to obtain some $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_n) \in \{0,1\}^{n\ell}$, which is sent to the verifier. Let $(\mathbf{x}_0, \mathbf{x}_1)$ be the vectors defined in the previous step, and B is defined in definition 5. The verifier accepts if

$$\mathbf{d} \in \mathbb{Y}_{0,\mathbf{x}_0} \cap \mathbb{Y}_{1,\mathbf{x}_1} \qquad \text{and} \qquad \bigoplus_{i=1}^{n} \langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \rangle = 0.$$

5.3 Analysis

First, we argue that the protocol is correct, i.e., the honest prover passes the tests with probability $1 - n^{-c}$, for some constant c. Observe that the verifier accepts at step 3 if $\mathbf{x}_0 \in \bar{\mathbf{X}}$ or $\mathbf{x}_1 \in \bar{\mathbf{X}}$. Since $\bar{\mathbf{X}}$ is a dense subset of X^n, it follows that:

- $\mathbf{x}_0 \in \bar{\mathbf{X}}$ or $\mathbf{x}_1 \in \bar{\mathbf{X}}$, and
- $(\mathbf{x}_0, \mathbf{x}_1) \in R_{\mathsf{ek}}$

except with inverse polynomial probability. Thus, the verifier rejects y with probability at most inverse polynomial. Conditioning on the verifier accepting in step 3, we have that the state of the prover equals

$$\frac{1}{\sqrt{2}} \left(|\mathbf{x}_0\rangle + |\mathbf{x}_1\rangle \right) |y\rangle$$

where $(\mathbf{x}_0, \mathbf{x}_1)$ are the pre-images of y under $f_{\mathsf{ek},0}$ and $f_{\mathsf{ek},1}$, respectively. On the one hand, measuring the first register in the computational basis returns a random pre-image of y, which allows the prover to pass the pre-image test with probability one, on the other hand, measuring the register in the Hadamard basis, returns a random vector orthogonal to $\mathsf{B}(\mathbf{x}_0) \oplus \mathsf{B}(\mathbf{x}_1)$, where B is the bit-decomposition operator. By definition, we have that

$$\langle \mathbf{d}, \mathsf{B}(\mathbf{x}_0) \oplus \mathsf{B}(\mathbf{x}_1) \rangle = \bigoplus_{i=1}^{n} \langle \mathbf{d}_i, \mathsf{B}_i(\mathbf{x}_0) \oplus \mathsf{B}_i(\mathbf{x}_1) \rangle = 0.$$

Furthermore, $\mathbf{d}$ belongs to the set $\mathbb{Y}_{0,\mathbf{x}_0} \cap \mathbb{Y}_{1,\mathbf{x}_1}$ with overwhelming probability. Thus the prover passes the equation test with probability negligibly close to one.

Next, we argue that the prover's state contains a qubit assuming that $\mathcal{F}$ is a wTCF family satisfying the adaptive hardcore bit property. The argument is essentially identical to the one shown in [Vid20] with minor syntactical modifications and we report it here only for completeness.

Theorem 2. *Let $\mathcal{F}$ be a wTCF family that satisfies the adaptive hardcore bit property. Let $|\psi\rangle$ be the state of a prover (after step 2) that succeeds with probability negligibly close to one. Then there exist two binary observables X and Z, such that $(|\psi\rangle, X, Z)$ is computationally close to a qubit. In particular, assuming the conjecture 2, the protocol can be instantiated based on the extended LHS assumption.*

Proof. Let $|\psi\rangle$ be the state of the prover after sending the y to the verifier. We assume without loss of generality that $|\psi\rangle$ is a pure bipartite state on $\mathcal{H}_{\mathsf{P}} \otimes \mathcal{H}_{\mathsf{M}}$, where the register P keeps the internal state of the prover. We also assume without loss of generality that the answer for $c = 0$ is obtained by measuring M on the computational basis. On the other hand, we also assume that the answer for $c = 1$ is obtained by computing $U|\psi\rangle$, for some unitary U, and measuring the resulting register M in the Hadamard basis. Next, we define the observables X and Z as

$$Z = \sum_{\mathbf{x} \in X^n} (-1)^{z_{\mathsf{ek},y}(\mathbf{x})} |\mathbf{x}\rangle\langle\mathbf{x}| \otimes \mathsf{Id}_{\mathsf{P}}$$

and

$$X = \sum_{\mathbf{d} \in \mathbb{Y}_{0,\mathbf{x}_0} \cap \mathbb{Y}_{1,\mathbf{x}_1}} (-1)^{x_{\mathsf{ek},y}(\mathbf{d})} U^\dagger (H_{\mathsf{M}}^{\otimes n\ell} \otimes \mathsf{Id}_{\mathsf{P}})^\dagger (|\mathbf{d}\rangle\langle\mathbf{d}|_{\mathsf{M}} \otimes \mathsf{Id}_{\mathsf{P}})(H_{\mathsf{M}}^{\otimes n\ell} \otimes \mathsf{Id}_{\mathsf{P}}) U$$

where the predicate $z_{\mathsf{ek},y}(\mathbf{x})$ labels as 0 the pre-image of y under $f_{\mathsf{ek},0}$ and as 1 the pre-image of y under $f_{\mathsf{ek},1}$ (other vectors are labeled arbitrarily). On the other hand, the predicate $x_{\mathsf{ek},y}(\mathbf{d})$ labels as 0 the $\mathbf{d}$ such that $\langle \mathbf{d}, \mathsf{B}(\mathbf{x}_0) \oplus \mathsf{B}(\mathbf{x}_1)\rangle = 0$ and as 1 all other vectors. We are now ready to show that $(|\psi\rangle, X, Z)$ is computationally close to a qubit. Let us rewrite

$$\begin{aligned}
&\frac{1}{4}\|\{X, Z\}|\psi\rangle\|^2 \\
&= \frac{1}{4}\|(XZ + ZX)|\psi\rangle\|^2 \\
&= \frac{1}{4}\langle\psi|(XZ + ZX)^\dagger(XZ + ZX)|\psi\rangle \\
&= \frac{1}{4}\langle\psi|(XZ + ZX)^2|\psi\rangle \\
&= \frac{1}{2}\left(\langle\psi|(XZ_0XZ_0)|\psi\rangle + \langle\psi|(XZ_1XZ_1)|\psi\rangle + \langle\psi|(Z_0XZ_0X)|\psi\rangle + \langle\psi|(Z_1XZ_1X)|\psi\rangle\right) \\
&= \langle\psi|(Z_0XZ_0)|\psi\rangle + \langle\psi|(Z_1XZ_1)|\psi\rangle + \mathrm{negl}
\end{aligned}$$

where the third equality uses that $(XZ + ZX)$ is Hermitian, the fourth equality follows from Lemma 10, and the last equality follows since we assume the prover to succeed with probability close to 1 and this $|\psi\rangle$ is negligibly close to an eigenstate of X with eigenvalue $+1$.

To complete the proof, we, therefore, need to show that the quantities $\langle\psi|\,(Z_0XZ_0)\,|\psi\rangle$ and $\langle\psi|\,(Z_1XZ_1)\,|\psi\rangle$ are negligible. We show this for the first term and the second case follows by symmetry. The proof consists of a reduction against the adaptive hardcore bit of property of $\mathcal{F}$: The reduction receives the key ek for the challenger and internally runs the prover to obtain the state $|\psi\rangle$ then it measures the register M in the computational basis to obtain some $\mathbf{x}$. If $\mathbf{x}$ is a pre-image of 1, then the reduction returns a $(\mathbf{x}, \mathbf{d})$, for a randomly sampled $\mathbf{d}$. Else, it applies the unitary U to $|\psi\rangle$ and measures the register M in the Hadamard basis to obtain $\mathbf{d}$. It returns $(\mathbf{x}, \mathbf{d})$.

In the former case ($\mathbf{x}$ being a pre-image of 1), we can lower bound the success probability of the reduction to be negligibly close to $1/2\,\langle\psi|\,Z_1\,|\psi\rangle$, since the prover is assumed to succeed with probability close to 1 and thus the post-measurement state is close to $Z_1\,|\psi\rangle$. Analogously, in the latter case ($\mathbf{x}$ being a pre-image of 0), the success probability of the reduction is negligibly close to

$$\langle\psi|\,Z_0X_0Z_0\,|\psi\rangle = 1/2(\langle\psi|\,Z_0\,|\psi\rangle + \langle\psi|\,Z_0XZ_0\,|\psi\rangle).$$

Overall, the success probability of the reduction is $1/2 + 1/2\,\langle\psi|\,Z_0XZ_0\,|\psi\rangle$. We can conclude that the second summand is negligible unless the reduction can break the adaptive hardcore bit property with a non-negligible probability. "In particular" part of theorem follows from Theorem 1.

To complete the proof, we need the following Lemma, which follows in verbatim from [Vid20].

Lemma 10. *Let X and Z be binary observables, then*

$$\frac{1}{2}(XZ + ZX)^2 = XZ_0XZ_0 + XZ_1XZ_1 + Z_0XZ_0X + Z_1XZ_1X.$$

Proof. Since X and Z are Hermitian and square to identity, we can rewrite

$$(XZ + ZX)^2 = 2\mathsf{Id} + XZXZ + ZXZX.$$

Recall that $Z = Z_0 - Z_1$, and thus we can expand

$$ZXZ = (Z_0 - Z_1)X(Z_0 - Z_1) = Z_0XZ_0 + Z_1XZ_1 - Z_0XZ_1 - Z_1XZ_0.$$

Using that $Z_0 + Z_1 = \mathsf{Id}$ we have

$$X = \mathsf{Id}X\mathsf{Id} = (Z_0 + Z_1)X(Z_0 + Z_1) = Z_0XZ_0 + Z_1XZ_1 + Z_0XZ_1 + Z_1XZ_0.$$

Combining the two equations above we obtain that $ZXZ = 2(Z_0XZ_0 + Z_1XZ_1) - X$. Plugging this into our first equation we obtain

$$\begin{aligned}(XZ + ZX)^2 &= 2\mathsf{Id} + XZXZ + ZXZX \\ &= 2\mathsf{Id} + X(2(Z_0XZ_0 + Z_1XZ_1) - X) + (2(Z_0XZ_0 + Z_1XZ_1) - X)X \\ &= 2\mathsf{Id} + 2XZ_0XZ_0 + 2XZ_1XZ_1 + 2Z_0XZ_0X + 2Z_1XZ_1X - 2X^2 \\ &= 2(XZ_0XZ_0 + XZ_1XZ_1 + Z_0XZ_0X + Z_1XZ_1X).\end{aligned}$$

Proof of Quantumness. We mention that our wTCF can be plugged into the work of [BKVV20][1] to obtain a classically verifiable proof of quantumness (PoQ). While PoQ is a *strictly weaker* goal than the qubit test that we described above, we explicitly mention this application since PoQ only requires the claw-freeness property. In particular, this means that we obtain a protocol for PoQ without the need to invoke conjecture 2 by relying only on the extended LHS assumption.

References

[AC02] Mark Adcock and Richard Cleve. A quantum goldreich-levin theorem with cryptographic applications. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 323–334. Springer, 2002.

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *ASIACRYPT 2020, Part II*, LNCS, pages 411–439. Springer, Heidelberg, December 2020.

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019.

[BKVV20] Zvika Brakerski, Venkata Koppula, Umesh V. Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In Steven T. Flammia, editor, *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 8:1–8:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[BM87] Richard P Brent and Brendan D McKay. Determinants and ranks of random matrices over zm. *Discrete Mathematics*, 66(1-2):35–49, 1987.

[BS20] Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, LNCS, pages 493–522. Springer, Heidelberg, May 2020.

[CGV22] Andrea Coladangelo, Shafi Goldwasser, and Umesh V. Vazirani. Deniable encryption in a quantum world. In *STOC 2022 (to appear)*, 2022.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018.

[FGK+10] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 279–295. Springer, Heidelberg, May 2010.

[GNW11] Oded Goldreich, Noam Nisan, and Avi Wigderson. On yao's xor-lemma. In Oded Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 273–301. Springer, 2011.

[1] Note that one needs to explicitly check for the domain membership of the preimages, similar to what is done for the qubit test protocol.

[GSV18] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In Mikkel Thorup, editor, *59th FOCS*, pages 956–966. IEEE Computer Society Press, October 2018.

[GV19] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In David Zuckerman, editor, *60th FOCS*, pages 1024–1033. IEEE Computer Society Press, November 2019.

[KCVY21] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. Classically-verifiable quantum advantage from a computational bell test. 2021.

[Mah18a] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018.

[Mah18b] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th FOCS*, pages 259–267. IEEE Computer Society Press, October 2018.

[NC02] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[Pei20] Chris Peikert. He gives C-sieves on the CSIDH. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, LNCS, pages 463–492. Springer, Heidelberg, May 2020.

[Vid20] Thomas Vidick. Course fsmp, fall'20: Interactions with quantum devices, 2020. `http://users.cms.caltech.edu/~vidick/teaching/fsmp/fsmp.pdf`.