How to Construct Quantum FHE, Generically

Aparna Gupte[*]
MIT
agupte@mit.edu

Vinod Vaikuntanathan[†]
MIT
vinodv@mit.edu

June 4, 2024

Abstract

We construct a (compact) quantum fully homomorphic encryption (QFHE) scheme starting from *any* (compact) classical fully homomorphic encryption scheme with decryption in NC^1, together with a dual-mode trapdoor function family. Compared to previous constructions (Mahadev, FOCS 2018; Brakerski, CRYPTO 2018) which made non-black-box use of similar underlying primitives, our construction provides a pathway to instantiations from different assumptions. Our construction uses the techniques of Dulek, Schaffner and Speelman (CRYPTO 2016) and shows how to make the client in their QFHE scheme classical using dual-mode trapdoor functions. As an additional contribution, we show a new instantiation of dual-mode trapdoor functions from group actions.

[*] Research supported by the Ida M. Green MIT Office of Graduate Education Fellowship and by the grants of the second author.

[†] Research supported in part by DARPA under Agreement Number HR00112020023, NSF CNS-2154149, a Simons Investigator award and a Thornton Family Faculty Research Innovation Fellowship from MIT.

Contents

1 Introduction

A quantum fully homomorphic encryption (QFHE) [Mah20] scheme enables a classical client with an input x to outsource the computation of a quantum circuit Q (on x) to an untrusted quantum server, while hiding x. The first QFHE scheme was constructed under the (super-polynomial) learning with errors (LWE) assumption in a groundbreaking work of Mahadev [Mah20]; this was followed shortly after by a work of Brakerski [Bra18] who, with a different construction, improved the assumption to polynomial LWE.[1]

The QFHE constructions of [Mah20, Bra18] built on classical FHE schemes that can be constructed from the LWE assumption. However, their schemes use very specific properties of specific FHE schemes. For example, [Mah20] used the dual-GSW FHE scheme [GSW13] together with a noisy trapdoor claw-free function built from LWE. The construction used the fact that the dual-GSW encryption of a bit b can be converted into the description of a claw-free function pair (f_0, f_1) that "encodes" the bit, in the sense that for two pre-images x_0 and x_1 such that $f_0(x_0) = f_1(x_1)$, the first bits of x_0 and x_1 xor to b. In a similar vein, [Bra18] crucially used the GSW encryption scheme and the discrete Gaussian structure of the set of all random strings consistent with a given ciphertext. In other words, both schemes used a particular instantiation of the underlying primitives and exploited their intricate interplay.

This raises a natural question which is the starting point of our work:

Can we generically transform any FHE scheme into a QFHE scheme?

Aside from the aesthetic appeal of such a result, it would give us a recipe to convert new classical FHE schemes, as they arise, into quantum FHE schemes; an example of a different FHE scheme is the IO-based FHE construction from [CLTV15].

1.1 Our Results

In this work, we make progress towards this goal by constructing a QFHE scheme starting with *any* (classical) FHE scheme (whose decryption circuit can be implemented in logarithmic depth) and *any* dual-mode trapdoor function (dTF) family.

We define a dual-mode trapdoor function family as follows: a dTF is a pair of functions (f_0, f_1) that either has the same image in one mode, or disjoint images in the other. Further, these two modes are computationally indistinguishable (see Section 3 for the formal definition). In contrast, a plain claw-free trapdoor function (as defined in [Mah18, BCM$^+$21]) is a pair of functions (f_0, f_1) for which it is computationally hard to find a claw, i.e. a pair of inputs (x_0, x_1) such that $f_0(x_0) = f_1(x_1)$.

Informal Theorem 1. *Let λ be the security parameter. Assuming the existence of a leveled fully homomorphic encryption scheme* HE *whose decryption algorithm can be implemented as a Boolean circuit with depth $O(\log \lambda)$, and a dual-mode trapdoor function family, there exists a quantum leveled fully homomorphic encryption scheme with a classical key generation procedure. The scheme encrypts classical information using classical ciphertexts.*

In other words, our result offers an answer to the open question above, except with two caveats: (1) we require the classical FHE scheme to have a shallow decryption circuit; and (2) we require an additional primitive, namely a dual-mode trapdoor function family. Both building blocks required for our

[1] In both cases, we get a leveled QFHE scheme, one that supports quantum circuits of an a-priori bounded depth, from LWE. To construct an unleveled QFHE scheme, both works rely on an appropriate circular security assumption, different from the one used to get an unleveled classical FHE scheme. From this point on, we focus on constructing a leveled QFHE scheme.

construction can be instantiated from the Ring LWE assumption giving us the first construction of QFHE from Ring LWE. We note that one could have come up with this instantiation by looking into the innards of [Mah20, Bra18]; however, with our construction, this instantiation is simply plug-and-play. Even with LWE-based constructions, any improvements in the underlying LWE-based FHE or LWE-based dual-mode trapdoor function translate directly into corresponding improvements in our QFHE scheme.

As an additional result, we show how to instantiate the dual-mode trapdoor function from group actions, building on the work of [AMR22] who construct a plain claw-free trapdoor function from the same assumption.

Informal Theorem 2. *There exists a dual-mode trapdoor function family under the extended linear hidden shift assumption (see [AMR22] and Definition 15) on group actions.*

It also turns out that we can construct an FHE scheme given an indistinguishability obfuscation (IO) scheme, together with a perfectly rerandomizable encryption scheme, following [CLTV15]. Together with a construction of a perfectly rerandomizable encryption scheme from group actions [Wic24], this gives us a QFHE scheme from (post-quantum, subexponentially secure) IO and group actions.

Informal Theorem 3. *Assuming that post-quantum sub-exponentially secure indistinguishability obfuscation exists and assuming the post-quantum hardness of the extended linear hidden shift assumption, there is a quantum leveled fully homomorphic encryption scheme with a classical key generation procedure. The scheme encrypts classical information using classical ciphertexts.*

We remark that we can extend this result to give unleveled QFHE using post-quantum IO.

1.2 Technical Overview

The starting point of our QFHE construction is the beautiful work of Dulek, Schaffner and Speelman [DSS16] which predated [Mah20], and showed how to build a protocol whereby a *semi-quantum* client with a (classical or quantum) input x can delegate to a fully quantum server with a quantum circuit Q the computation of $Q(x)$. In addition to encrypting the input x, the client also generates polynomially many copies of a *quantum gadget* that helps the server do the quantum homomorphic evaluation of Q on the encrypted x.

In a nutshell, the key idea in our work is to come up with a mechanism for the client to outsource to the server the preparation of these quantum gadgets, using a dual-mode trapdoor function. This results in a completely classical client, and consequently a QFHE scheme.[2] We now proceed to describe the construction in more detail, starting with the DSS protocol.

The DSS scheme builds on the work of Broadbent and Jeffery [BJ15] and encrypts quantum states (or classical strings) with a quantum one-time pad. That is, a single-qubit state $|\psi\rangle$ is encrypted as $\mathsf{X}^x\mathsf{Z}^z\,|\psi\rangle$ where $x, z \leftarrow \{0,1\}$ are uniformly random bits and

$$\mathsf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathsf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

are the Pauli X and Z operators. Additionally, the ciphertext also contains a (classical) homomorphic encryption of the one-time pad keys (x, z).

[2] This is, in effect, a remote state preparation protocol for the DSS gadget states; however, in the QFHE setting (as opposed to the verification setting), the protocol only needs to be secure against a semi-honest/specious server.

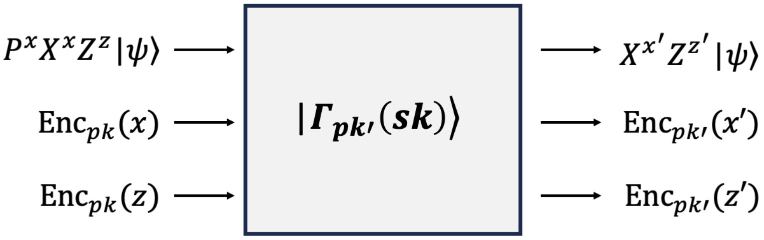


Figure 1: The functionality of the DSS quantum gadget $|\Gamma(sk)\rangle$: take as input a state $P^x X^x Z^z |\psi\rangle$ as well as encryptions of x and z, and produce as output $X^{x'} Z^{z'} |\psi\rangle$ together with encryptions of the new one-time pad keys x' and z'.

We recall that any quantum circuit can be implemented using Clifford gates and the single-qubit T gate

$$\mathsf{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Homomorphic evaluation of Clifford gates on states encrypted using the Pauli one-time pad is easy as Clifford gates nicely "commute" through the Pauli one-time pad. For example,

$$\mathsf{H}\mathsf{X}^x\mathsf{Z}^z |\psi\rangle = \mathsf{Z}^x\mathsf{X}^z\mathsf{H} |\psi\rangle$$

All that needs to be done is to homomorphically update encryption of the one-time pads. When the non-Clifford T gate is applied to a one-time padded state, we end up with an extra non-Pauli error in the form of a P gate,

$$\mathsf{T}\mathsf{X}^x\mathsf{Z}^z |\psi\rangle = \mathsf{P}^x\mathsf{X}^x\mathsf{Z}^z\mathsf{T} |\psi\rangle .$$

The natural solution would be to apply a $\mathsf{P}^\dagger$ gate to this state if $x = 1$, and otherwise do nothing. However, the evaluator only has access to an encryption $\tilde{x}$ of the one-time pad key x. Dulek, Schaffner and Speelman [DSS16] solve this issue by writing the decryption algorithm (with the secret key sk hardcoded) as a branching program and converting it into a quantum "teleportation gadget" $\Gamma(sk)$ which the client prepares during key generation. We henceforth refer to this gadget as the DSS teleportation gadget. The server then teleports the qubit $\mathsf{P}^x\mathsf{X}^x\mathsf{Z}^z\mathsf{T} |\psi\rangle$ through the gadget in a way that depends on $\tilde{x}$, and the gadget implicitly decrypts $\tilde{x}$ and applies the $(\mathsf{P}^\dagger)^x$ correction to the qubit. The functionality achieved by the DSS gadget is depicted in Figure 1. For completeness, we describe how the DSS gadget looks and how it is used to achieve the desired functionality in Appendix A.

At a high level, the DSS gadget consists of Bell pairs that are wired in a way that depends on the bits of the secret key sk used to decrypt the ciphertext $\mathsf{Enc}_{pk}(x)$. The key new contribution of our work is a method that allows a completely classical client to delegate the preparation of the DSS teleportation gadget to the server. At the center of our method is a procedure that prepares a "hidden" Bell pair. The procedure takes three qubits in registers $0, 1, 2$ in the product state $|+\rangle_0 \otimes |+\rangle_1 \otimes |+\rangle_2$, and a particular kind of "encryption" of a bit μ. The procedure then outputs a (one-time padded) state where the qubits on registers μ and 2 wired together to form a maximally entangled Bell state that is in tensor product with the qubit $|+\rangle_{1-\mu}$ in the $1-\mu$ register,

$$\left|\Phi^+\right\rangle_{\mu,2} \otimes |+\rangle_{1-\mu} .$$

We do this with the help of a "4-to-2" dual-mode trapdoor function (which we can obtain generically from dual-mode trapdoor functions, see Section 3.1), a new primitive we define as follows (see Figure 3 for a pictorial illustration): a tuple of four functions $f_{b_1,b_2} : \mathcal{X} \to \mathcal{Y}$ for b_1, b_2 such that depending on a "mode" bit $\mu \in \{0,1\}$, either

1. (Case $\mu = 0$) two functions f_{b_1,b_2} and $f_{b_1',b_2'}$ have the same image if $b_1 = b_1'$ and otherwise they have disjoint images, or
2. (Case $\mu = 1$) two functions f_{b_1,b_2} and $f_{b_1',b_2'}$ have the same image if $b_2 = b_2'$. Otherwise they have disjoint images.

Further, without a trapdoor, no efficient adversary should be able to guess μ significantly better than randomly guessing. Such a tuple of functions encoding the mode bit μ is the specific "encryption" we use to prepare the required hidden Bell pairs. Given such a family of functions, we prepare the state

$$\frac{1}{2\sqrt{2|\mathcal{X}|}} \sum_{\substack{u,v,w\in\{0,1\}\\ x\in\mathcal{X}}} |u,v,w\rangle_{0,1,2} \otimes |x\rangle_3 \otimes |f_{u\oplus w,v\oplus w}(x)\rangle_4 .$$

Suppose for simplicity that $\mu = 0$. Measuring the fourth register to get some $y \in \mathcal{Y}$ collapses the state to preimages under either the $b_1 = 0$ pair $(f_{0,b_2})_{b_2\in\{0,1\}}$ or the $b_1 = 1$ pair $(f_{1,b_2})_{b_2\in\{0,1\}}$. Again, for simplicity, let us consider the branch where $b_1 = 0$. The state we are now left with is

$$\frac{1}{2}\left(|000\rangle|x_{0,0}\rangle + |101\rangle|x_{0,1}\rangle + |111\rangle|x_{0,0}\rangle + |010\rangle|x_{0,1}\rangle\right)_{0,1,2,3} ,$$

where x_{b_1,b_2} is such that $f_{b_1,b_2}(x_{b_1,b_2}) = y$. Now, we measure the register 3 (containing the preimages) in the Hadamard basis, so that we obtain a string d and the state (up to global phase)

$$\begin{aligned}&\frac{1}{2}\left(|000\rangle + (-1)^z|101\rangle + |111\rangle + (-1)^z|010\rangle\right)\\ &\qquad = \frac{1}{2}(\mathsf{Z}^z \otimes \mathsf{Z}^z \otimes \mathsf{I})\left|\Phi^+\right\rangle_{0,2} \otimes |+\rangle_1 ,\end{aligned}$$

where $z = d \cdot (x_{0,0} \oplus x_{0,1})$.

This basic protocol allows us to dequantize the DSS client in the following manner: depending on the secret key sk, the client generates classical evaluation keys for the 4-to-2 dual-mode trapdoor functions (where the modes depend on the bits of sk) and sends these to the server. The server then performs the above procedure for all the qubits in the gadget, thereby wiring the Bell pairs correctly depending on the bits of sk. Once the server has prepared this gadget $\Gamma(sk)$, it can simply follow the DSS scheme for evaluating the T gates and correcting the P errors.

1.3 Related Work

Quantum Fully-Homomorphic Encryption. Broadbent and Jeffery [BJ15] gave the first QFHE scheme assuming only a classical FHE scheme, although they considered the setting of a quantum client. They constructed a (non-compact) scheme where the ciphertext size grows polynomially with the number of non-Clifford gates in the circuit. They also gave a different quantum FHE scheme in which the quantum client produces quantum evaluation keys, the size of which grows doubly exponentially in the non-Clifford gate-depth. Dulek, Schaffner and Speelman [DSS16] improved on this and gave a QFHE scheme, also for a quantum client, but with the size of the quantum evaluation key growing polynomially with the non-Clifford gate-depth. However, they required the underlying classical FHE scheme to have a decryption circuit in NC^1.

The work of Mahadev [Mah20] was a major breakthrough and constructed the first QFHE scheme for a completely classical client, under the quantum hardness of LWE (with superpolynomial modulus-to-noise ratio). Brakerski [Bra18] later improved the parameters, and gave a different QFHE scheme for a classical client under a weaker assumption, namely LWE with a polynomial modulus-to-noise ratio.

Remote State Preparation. The main conceptual idea of our work is to give a protocol by which the server can blindly prepare the quantum evaluation keys of [DSS16], given only classical messages from the client and therefore side-stepping the need for a quantum client as well as quantum communication. This idea is inspired by the work on remote state preparation, a protocol that allows a quantum server to blindly prepare a quantum state specified by the classical client. The notion of remote state preparation was first introduced by [DK16], and subsequently studied by [GV19, GMP22]. We note that remote state preparation usually refers to a stronger variant in the setting of a malicious server, where the protocol to prepare the quantum state is not only *blind*, but also *verifiable* by the classical client. However in the QFHE setting, the server is semi-honest, and we only require the weaker notion of blind (but not necessarily verifiable) remote state preparation.

Organization of the Paper. In Section 2, we introduce some notation and define (quantum) homomorphic encryption. In Section 3, we define dual-mode trapdoor functions (dTFs), and show how to construct a variant we call "4-to-2" dual-mode trapdoor functions from plain dual-mode trapdoor functions. We also give an amplification lemma that will later allow us to construct dTFs from group actions with negligible correctness error. In Section 4 we use this primitive to show that we can dequantize the DSS client, and we present our QFHE scheme in Section 5. Section 6 lists the known instantiations of the two primitives we require, namely classical FHE and dual-mode trapdoor functions, from several cryptographic assumptions. Finally, in Section 7, we show an instantiation based on hardness assumptions related to group actions.

2 Preliminaries

Notation. Following the convention in [DSS16], we denote ciphertexts with a tilde sign, when the encryption scheme and the public/private key are clear from the context. For example $\tilde{a}$ denotes an encryption of a. When a sequence of keys is involved, we write $\tilde{a}^{[i]}$ to denote an encryption of a under the i-th key in a set. If $\mathcal{M}$ is a quantum register, we denote the set of density operators on $\mathcal{M}$ by $D(\mathcal{M})$. We denote the trace distance between density matrices ρ and σ as $T(\rho, \sigma)$, and the total variation distance between probability distribution D_1, D_2 as $\mathrm{TV}(D_1, D_2)$.

Definition 1 (Hellinger Distance). *For two probability density functions f, g over a finite domain $\mathcal{X}$, the squared Hellinger distance between f and g is defined as*

$$H^2(f,g) = \frac{1}{2}\sum_{x\in\mathcal{X}}\left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 = 1 - \sum_{x\in\mathcal{X}}\sqrt{f(x)g(x)}.$$

Lemma 1. *Let D_1, D_2 be two probability density functions over a finite domain $\mathcal{X}$, then*

$$\frac{1}{2}H^2(D_1,D_2) \leq TV(D_1,D_2) \leq H(D_1,D_2).$$

Lemma 2. *Let $\mathcal{X}$ be a finite set and let D_1, D_2 be probability densities on $\mathcal{X}$. Let*

$$|\psi_1\rangle = \sum_{x\in\mathcal{X}}\sqrt{D_1(x)}\,|x\rangle \quad \textit{and} \quad |\psi_2\rangle = \sum_{x\in\mathcal{X}}\sqrt{D_2(x)}\,|x\rangle \ .$$

Then, the trace distance between the pure states $|\psi_1\rangle, |\psi_2\rangle$ is

$$T(|\psi_1\rangle\langle\psi_1|, |\psi_2\rangle\langle\psi_2|) = \sqrt{1-(1-H^2(D_1,D_2))^2}.$$

2.1 Homomorphic Encryption

Definition 2 (Classical Homomorphic encryption scheme). *Let λ be the security parameter. A homomorphic encryption scheme* HE *is a tuple of p.p.t. algorithms* $\mathsf{HE} = (\mathsf{HE.KeyGen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ *with the following properties.*

- *Key Generation. The probabilistic key generation algorithm $(pk, evk, sk) \leftarrow \mathsf{HE.KeyGen}(1^\lambda)$ outputs a public key pk, an evaluation evk and a secret key sk.*
- *Encryption. The probabilistic encryption algorithm $c \leftarrow \mathsf{HE.Enc}_{pk}(x)$ uses the public key pk and encrypts a single bit message $x \in \{0,1\}$ into a ciphertext c.*
- *Decryption. The deterministic decryption algorithm $x \leftarrow \mathsf{HE.Dec}_{sk}(c)$ uses the secret key sk and decrypts a ciphertext c to recover the message $x \in \{0,1\}$.*
- *Evaluation. The (deterministic or probabilistic) evaluation algorithm $c' \leftarrow \mathsf{HE.Eval}_{evk}(\mathcal{C}, (c_1,\ldots,c_\ell))$ uses the evaluation key evk, takes as input a circuit $\mathcal{C} : \{0,1\}^\ell \rightarrow \{0,1\}$ and sequence of ciphertexts $c_1,\ldots,c_\ell$, and outputs a ciphertext c'.*

We often overload the functionality of the encryption and decryption procedures by allowing them to take in multi-bit messages as inputs, and produce a sequence of ciphertexts that correspond to bit-by-bit encryptions.

Definition 3 (Full homomorphism and compactness). *A scheme* HE *is fully homomorphic, if for any efficiently computable circuit $\mathcal{C}$ and any set of inputs $x_1, x_2, \ldots, x_\ell$,*

$$\Pr[\mathsf{HE.Dec}_{sk}(c') \neq \mathcal{C}(x_1,\ldots,x_\ell)] = \mathsf{negl}(\lambda).$$

where the probability is taken over $(pk, evk, sk) \leftarrow \mathsf{HE.KeyGen}(1^\lambda)$ and $c_i \leftarrow \mathsf{HE.Enc}_{pk}(x_i)$, and $c' \leftarrow$ $\mathsf{HE.Eval}_{evk}(\mathcal{C}, (c_1,\ldots,c_\ell))$.

A fully homomorphic encryption scheme is compact if its decryption circuit is independent of the evaluated function. The scheme is leveled fully homomorphic if it takes 1^L as additional input in key generation, and can only evaluate depth L Boolean circuits.

2.2 Quantum Homomorphic Encryption

We now give a definition for quantum homomorphic encryption based on the definition given in [BJ15]. In this work, we focus on quantum homomorphic encryption schemes with entirely classical key generation procedures.

Definition 4 (Quantum Homomorphic Encryption)**.** *A quantum homomorphic encryptions scheme is a* QHE *is a tuple of quantum polynomial-time algorithms* $(\mathsf{QHE.KeyGen}, \mathsf{QHE.Eval}, \mathsf{QHE.Dec})$*:*

- *Key Generation. The probabilistic key generation algorithm* $(pk, sk, evk) \leftarrow \mathsf{QHE.KeyGen}(1^\lambda)$ *takes a unary representation of the security parameter as input and outputs a classical public key* pk*, a classical secret key* sk *and a classical evaluation key* evk*.*
- *Encryption. For every possible value of* pk*, the quantum channel* $\mathsf{QHE.Enc}_{pk} : D(\mathcal{M}) \to D(\mathcal{C})$ *maps a state in the message space* $\mathcal{M}$ *to a state (the cipherstate) in the cipherspace* $\mathcal{C}$*.*
- *Decryption. For every possible value of* sk*,* $\mathsf{QHE.Dec}_{sk} : D(\mathcal{C}') \to \mathcal{M}$ *is a quantum channel that maps the state in* $D(\mathcal{C}')$ *to a quantum state in* $D(\mathcal{M})$*.*
- *Homomorphic Circuit Evaluation. For every quantum circuit* C*, with induced channel* $\Phi_{\mathsf{C}} : D(\mathcal{M}^{\otimes n}) \to D(\mathcal{M}^{\otimes m})$*, we define a channel* $\mathsf{QHE.Eval}_{evk}(\mathsf{C}, \cdot) : D(\mathcal{C}^{\otimes n}) \to D(\mathcal{C}'^{\otimes m})$ *that maps an* n*-fold cipherstate to an* m*-fold cipherstate.*

Definition 5 (Quantum Full Homomorphism and Compactness)**.** *Let* λ *be the security parameter. A quantum homomorphic encryption scheme* QHE *is fully homomorphic, if for any efficiently computable quantum circuit* C *with induced channels* $\Phi_{\mathsf{C}} : \mathcal{M}^{\otimes n(\lambda)} \to \mathcal{M}^{\otimes m(\lambda)}$*, and for any input* $\rho \in D(\mathcal{M}^{\otimes n(\lambda)} \otimes \mathcal{E})$*, there exists a negligible function* negl *s.t. for* $(pk, sk, evk) \leftarrow \mathsf{QHE.KeyGen}(1^\lambda)$*, the state*

$$\mathsf{QHE.Dec}_{sk}^{\otimes m(\lambda)}\left(\mathsf{QHE.Eval}_{evk}(\mathsf{C}, \mathsf{QHE.Enc}_{pk}^{n(\lambda)}(\rho))\right)$$

is at most $\mathsf{negl}(\lambda)$*-away in trace distance from the state* $\Phi_{\mathsf{C}}(\rho)$*.*

A quantum homomorphic encryption is compact if its decryption circuit is independent of the evaluated circuit.

The scheme is leveled fully homomorphic if it takes 1^L *as additional input in key generation, and can only evaluate circuits of* T*-depth at most* L*.*

Definition 6 (The Quantum IND-CPA Game [BJ15])**.** *Let* $\mathcal{M}, \mathcal{S}, \mathcal{C}$ *be registers. The quantum CPA indistinguishability experiment with respect to a quantum homomorphic encryption scheme* S *and a quantum polynomial-time adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *with security parameter* λ*, denoted as* $\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}, \mathcal{A} \rangle}(1^\lambda)$ *is defined as follows.*

1. *The challenger runs* $\mathsf{S.Gen}(1^\lambda)$ *to obtain keys* (pk, sk, evk)*.*
2. *Adversary* $\mathcal{A}_1$ *is given* (pk, evk) *and outputs a quantum state* $\rho_{\mathcal{M},\mathcal{S}}$ *on* $\mathcal{M} \otimes \mathcal{S}$*.*
3. *The challenger samples a uniformly random bit* $r \leftarrow \{0,1\}$*. If* $r = 0$*, the challenger outputs in* $\mathcal{C}$ *an encryption* $\mathsf{S.Enc}_{pk}(|0\rangle\langle 0|)$ *of the state* $|0\rangle\langle 0|$*. If* $r = 1$*, the challenger outputs in* $\mathcal{C}$ *an encryption* $\mathsf{S.Enc}_{pk}(\rho_{\mathcal{M}})$ *of the state in* $\mathcal{M}$*.*
4. *Adversary* $\mathcal{A}_2$ *obtains* $\mathcal{C} \otimes \mathcal{S}$ *and outputs a bit* r'*.*

5. *If $r = r'$, the adversary $\mathcal{A}$ wins, and the outcome of the experiment is defined to be 1. Otherwise, the outcome of the experiment is defined to be 1.*

Definition 7 (Quantum CPA Indistinguishability [BJ15])**.** *A quantum homomorphic encryption scheme S is quantum-IND-CPA secure if there is no quantum polynomial-time adversary wins the quantum IND-CPA game with more than negligible advantage. That is, for any quantum polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{G}^{q\text{-}CPA}_{\langle \mathsf{S}, \mathcal{A} \rangle}(1^\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Broadbent and Jeffery [BJ15] show that the above security definition is equivalent to security w.r.t. a quantum IND-CPA game where the adversary sends multiple messages to the challenger, and the challenger either encrypts all of them or encrypts a sequence of all $|0\rangle \langle 0|$ states.

2.3 Discrete Gaussians

Definition 8 (Truncated Discrete Gaussian)**.** *Let $B > 0$ be a real number and let q be a positive integer. The truncated discrete Gaussian distribution over $\mathbb{Z}_q$ with parameter B is the distribution supported on $\{x \in \mathbb{Z}_q : \|x\| \leq B\}$ with density proportional to*

$$\rho_{\mathbb{Z}_q, B}(x) \propto \exp\left(\frac{-\pi \|x\|^2}{B^2}\right) .$$

For a positive integer m the truncated discrete Gaussian distribution on $\mathbb{Z}_q^m$ with parameter B is the distribution supported on $\{\mathbf{x} \in \mathbb{Z}_q : \|x\| \leq B\sqrt{m}\}$ with density

$$\rho_{\mathbb{Z}_q^m, B}(\mathbf{x}) = \rho_{\mathbb{Z}_q, B}(x_1) \cdots \rho_{\mathbb{Z}_q, B}(x_m).$$

Lemma 3. *Let $B > 0$ be a real number and let q, m be positive integers. Consider $\mathbf{e} \in \mathbb{Z}_q^m$ such that $\|\mathbf{e}\| \leq B\sqrt{m}$. The Hellinger distance between the distribution $\rho = \rho_{\mathbb{Z}_q^m, B}$ and the shifted distribution $\rho + \mathbf{e}$ with density $(\rho + \mathbf{e})(\mathbf{x}) = \rho(\mathbf{x} - \mathbf{e})$ satisfies*

$$H^2(\rho, \rho + \mathbf{e}) \leq 1 - \exp\left(\frac{-2\pi\sqrt{m}\|\mathbf{e}\|}{B}\right) \leq \frac{2\pi\sqrt{m}\|\mathbf{e}\|}{B} .$$

2.4 Learning With Errors

Definition 9 (The $\mathrm{LWE}_{m,n,q,\chi}$ problem)**.** *Let λ be the security parameter, let m, n, q be integer functions of λ. Let $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}$. The $LWE_{m,n,q,\chi}$ problem is to distinguish between the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q)$ and $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$.*

The problem where m can be an arbitrary polynomially bounded function of $n \log q$ is written as $LWE_{n,q,\chi}$.

Definition 10 (The $\mathrm{LWE}_{n,q,\chi}$ Assumption)**.** *There is no quantum polynomial-time proceudre that solves the $LWE_{n,q,\chi}$ problem with advantage more than negligible in λ.*

There is evidence to believe that the $\mathrm{LWE}_{n,q,\rho_{\mathbb{Z}_q,\sigma}}$ assumption is true for certain parameter regimes, where $\sigma = \alpha q \geq 2\sqrt{n}$, and α is the noise-to-modulus ratio. In particular, [Reg09] and [PRSD17] show that a quantum polynomial-time algorithm for $\mathrm{LWE}_{n,q,\rho_{\mathbb{Z}_q,\sigma}}$ would give a quantum polynomial-time algorithm for SIVP (shortest independent vectors problem) in for n-dimensional lattices within an approximation factor of $\gamma = \tilde{O}(n/\alpha)$. The best known (classical or quantum) algorithms for SIVP run in time $2^{\tilde{O}(n/\log\gamma)}$, which is efficient for $\gamma = 2^{\tilde{O}(n/\log n)}$. If SIVP is hard for a (smaller) superpolynomial approximation factor γ, LWE is hard for a corresponding inverse superpolynomial noise-to-modulus ratio.

Theorem 1 (Theorem 5.1 in [MP12])**.** *Let $n, m \geq 1$ and $q \geq 2$ be such that $m = \Omega(n \log q)$. There is an efficient randomized algorithm $\mathsf{GenTrap}(1^n, t^m, q)$ that returns a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and a trapdoor $t_{\mathbf{A}}$ such that the distribution of $\mathbf{A}$ is negligibly (in n) close to the uniform distribution. Moreover, there is an efficient algorithm Invert that, on input $\mathbf{A}, t_{\mathbf{A}}$ and $\mathbf{As}+\mathbf{e}$ where $\|\mathbf{e}\|_2 \leq q/(C\sqrt{n\log q})$ and C is a universal constant, returns $\mathbf{s}$ and $\mathbf{e}$ with overwhelming probability over $(\mathbf{A}, t_{\mathbf{A}}) \leftarrow \mathsf{GenTrap}(1^n, t^m, q)$.*

3 Dual-mode Trapdoor Functions

A central primitive we make use of in our QFHE scheme is what we will call a dual-mode trapdoor function (dTF) family (and a variant, 4-to-2 dTFs, defined in Section 3.1). This primitive is a slight variant of noisy trapdoor claw-free function families (NTCFs) and *extended* NTCFs, which find several applications in quantum cryptography, starting with the breakthrough schemes for QFHE and classical verification of BQP computations of Mahadev [Mah20, Mah18].[3]

Informally, a dTF family is a collection of pairs of functions f_0, f_1 that have either disjoint images (disjoint mode $\mu = 0$) or they the same image (lossy mode $\mu = 1$). We note that in reality the lossy mode property is slightly more technical to accommodate non-injective functions. Further, given the evaluation keys for f_0, f_1, it should be computationally hard to tell in which mode it is. The keys are sampled along with trapdoors that allow efficient inversion, and therefore also distinguish the two modes. See Figure 2 for intuition.

Section 6 describes constructions of dTFs from LWE and group actions. We note that none of these constructions give us "clean" dTFs; each assumption gives rise to a different notion of noise/error that deviates from the natural ideal definition. To capture both these different notions of noise, we introduce a probability distribution over the domain $\mathcal{X}$, and an correctness-error parameter ε in our definition. Although the construction of dTFs from group actions gives us inverse-polynomial correctness error at first, we show an amplification lemma in Section 3.2 that allows us to obtain negligible correctness error.

Definition 11 (ε-Weak Dual-Mode Trapdoor Function Family)**.** *Let λ be the security parameter, and let $\mathcal{X}$, $\mathcal{Y}$ be finite sets.*[4] *Let $\mathcal{K}$ be a finite set of keys. A family of functions*

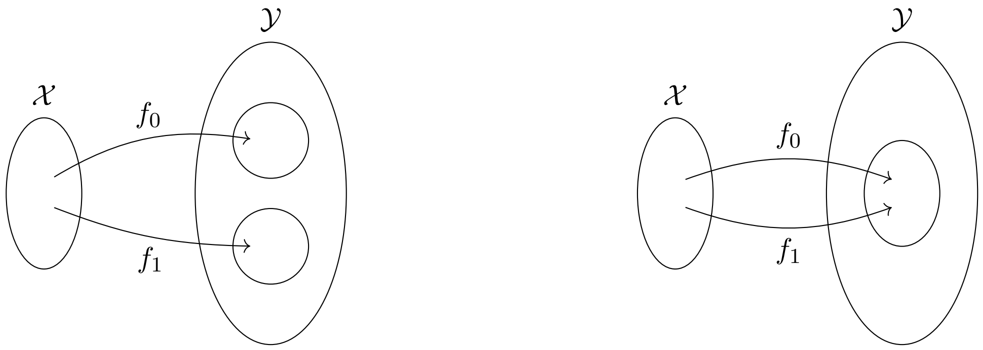
$$\mathcal{F} = \{f_{k,b} : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K}, b\in\{0,1\}}$$

*is called a ε-**weak dual-mode trapdoor function family** if there exists an associated a family of probability distributions $\mathcal{D} = \{D_{k,b}\}_{k\in\mathcal{K},b\in\{0,1\}}$ over $\mathcal{X}$ such that the following conditions hold.*

1. ***Efficient key generation.*** *There is an efficient probabilistic algorithm $\mathsf{Gen}_{\mathcal{F}}$ which takes as input the security parameter 1^λ and a "mode" bit $\mu \in \{0, 1\}$, and generates a key $k(\mu) \in \mathcal{K}$ together with a*

[3] Our dTFs are a weakening of Mahadev's extended NTCFs in two ways: we do not require the adaptive hardcore bit property, and we do not need the functions to be injective. Relaxing the definition to allow many-to-one functions allows us to obtain dTFs with good correctness properties from group actions, as well as dTFs from weaker LWE assumption based on [Bra18].

[4] We assume that there is an efficient binary encoding that maps elements $x \in \mathcal{X}$ to binary strings of length t, say. For simplicity, we do not use additional notation to distinguish between a set element $x \in \mathcal{X}$ and its binary representation. We write $\bar{\mathcal{X}}$ to denote $\{0,1\}^t$, and when we write $d \cdot x$ for $d \in \bar{\mathcal{X}}$, we think of x as its binary encoding, and the dot product as being over $\mathbb{F}_2$.



(a) $\mu = 0$

(b) $\mu = 1$

Figure 2: Dual-mode trapdoor functions (dTFs): In mode $\mu = 0$, the functions f_0 and f_1 have disjoint images, and in mode $\mu = 1$ they have the same image. The two modes are computationally indistinguishable.

trapdoor t_k,

$$(k(\mu), t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu).$$

2. ***Efficient function evaluation.*** *There is an efficient deterministic algorithm that takes as input any $k \in \mathcal{K}$, $b \in \{0,1\}$, and $x \in \mathcal{X}$ and computes $f_{k,b}(x)$.*

3. ***Efficient state preparation.*** *There is a quantum polynomial-time algorithm that prepares, for every $k \in \mathcal{K}, b \in \{0,1\}$, a state negligibly close in trace distance to the following state:*

$$\sum_{x \in \mathcal{X}} \sqrt{D_{k,b}(x)} \, |x\rangle \, .$$

4. ***Efficient Partial Inversion and Phase Computation with the Trapdoor.*** *There exists an efficient deterministic algorithm that takes as input t_k, y and outputs the set*

$$\{b \in \{0,1\} : \exists x \in \mathcal{X} \text{ s.t. } f_{k,b}(x) = y\}.$$

Additionally, there exists another efficient deterministic algorithm that takes as input $y \in \mathcal{Y}$, $d \in \bar{\mathcal{X}}$ and $b \in \{0,1\}$, along with trapdoor t_k, and computes (up to a common normalization factor) $\alpha_{k,y,d}(b)$, defined as

$$\alpha_{k,y,d}(b) := \sum_{x \in \mathcal{X}} (-1)^{d \cdot x} \sqrt{D_{k,b}(x)} \, .$$

When k, y and d are clear from the context, we will sometimes abbreviate $\alpha_{k,y,d}(b)$ as α_b.

5. ***Dual-mode.*** *Intuitively, we want the images of functions $f_{k,0}$ and $f_{k,1}$ to be either disjoint ($\mu = 0$) or nearly identical ($\mu = 1$), depending on the mode μ. More precisely, for any $\mu \in \{0,1\}$, for all but a negligible fraction of keys $(k(\mu), \cdot) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu)$,*

   (a) Disjoint mode ($\mu = 0$): For $x \leftarrow D_{k,0}$, $x' \leftarrow D_{k,1}$ we have that $f_{k,0}(x) \neq f_{k,1}(x')$ with probability 1.

   (b) *Lossy mode ($\mu = 1$): for $b \leftarrow \{0,1\}$, $x \leftarrow D_{k,b}$, $y = f_{k,b}(x)$, with probability $1 - \varepsilon$ there exists some $s \in \{0,1\}$ such that $\alpha_{k,y,d}(0) = (-1)^s \cdot \alpha_{k,y,d}(1)$ for all $d \in \bar{\mathcal{X}}$.*

6. ***Mode Indistinguishability.** Given a key $k(\mu)$ generated with $\mathsf{Gen}(1^\lambda, \mu)$, no efficient algorithm can guess μ with greater than negligible advantage. Formally, for all quantum polynomial-time procedures $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that*

$$\left| \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda,0)}[\mathcal{A}(k)] - \Pr_{(k,t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda,1)}[\mathcal{A}(k)] \right| \leq \mathsf{negl}(\lambda).$$

Remark 1. *If ε is negligible in the security parameter, we simply say the function family is a dual-mode trapdoor function family.*

For our amplification lemma (Section 3.2), it will be convenient for us to define a stronger notion of dTFs, where the functions are injective and efficiently invertible with a trapdoor.

Definition 12. *We additionally call an ε-weak dTF family $\mathcal{F}$ **injective and invertible** if the following properties hold respectively.*

1. *For all $k \in \mathcal{K}$, $b \in \{0,1\}$, $f_{k,b}$ is injective on the support of $D_{k,b}$. That is, for all $x \neq x' \in \mathrm{support}(D_{k,b})$, we have that $f_{k,b}(x) \neq f_{k,b}(x')$.*

2. *There exists an efficient deterministic algorithm $\mathsf{Inv}_{\mathcal{F}}$ such that for all $y \in \mathcal{Y}$ and key-trapdoor pairs (k, t_k) generated by $\mathsf{Gen}_{\mathcal{F}}$,*

$$\{(b,x) \mid b \in \{0,1\}, x \in \mathcal{X}, y = f_{k,b}(x)\} \leftarrow \mathsf{Inv}_{\mathcal{F}}(t_k, y).$$

3. *For every $b, b' \in \{0,1\}$ and $x, x' \in \mathcal{X}$, $f_{k,b}(x) = f_{k,b'}(x')$ implies that $D_{k,b}(x) = D_{k,b'}(x')$.*

3.1 4-to-2 dTFs and their construction from dTFs

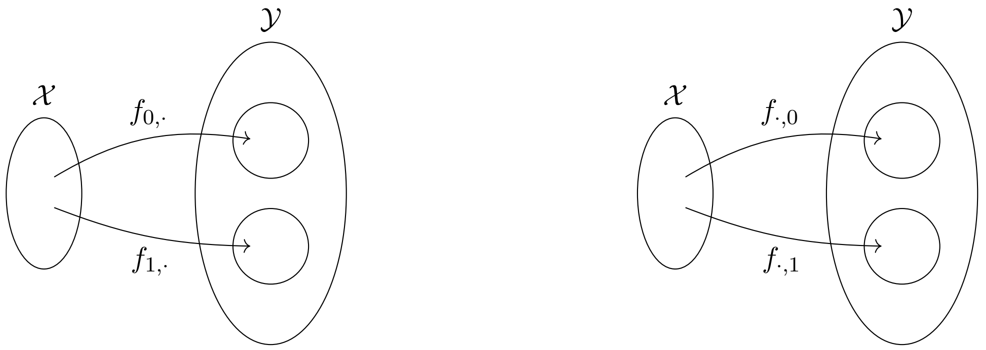
In this subsection, we define 4-to-2 dTFs and give a simple transformation that takes a dTF family and outputs a 4-to-2 dTF family. We define a 4-to-2 dual-mode trapdoor function to be a family of 4-tuples of functions f_{b_1,b_2} indexed by $b_1, b_2 \in \{0,1\}$. In mode $\mu = 0$, functions $f_{0,0}$ and $f_{0,1}$ have the same image, as do functions $f_{1,0}$ and $f_{1,1}$. Further, these two images are disjoint. Symmetrically, in mode $\mu = 1$, functions $f_{0,0}$ and $f_{1,0}$ share an image, as do functions $f_{0,1}$ and $f_{1,1}$. The intuition for this primitive is best conveyed through a picture (see Figure 3).[5]

Definition 13 (ε-weak 4-to-2 dual-mode trapdoor functions)**.** *Let λ be the security parameter, and let $\mathcal{X}$, $\mathcal{Y}$ be finite sets. Let $\mathcal{K}$ be a finite set of keys. A family of functions*

$$\mathcal{F} = \{f_{k,b_1,b_2} : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}, b_1, b_2 \in \{0,1\}}$$

*is a ε**-weak** 4**-to-**2 **dual-mode trapdoor function family** if there exists a family of probability distributions $\mathcal{D} = \{D_{k,b_1,b_2}\}_{k \in \mathcal{K}, b_1, b_2 \in \{0,1\}}$ over $\mathcal{X}$ such that the following conditions hold.*

[5] We remark that there is a natural way to generalize dTFs and 4-to-2 dTFs so that they are specializations of the same primitive. In short, we can define a k-mode trapdoor function family to be a collection of tuples of injective functions $(f_i)_{i \in \mathcal{I}}$ with finite index set $\mathcal{I}$. The two modes are defined by two distinct partition functions $p_0, p_1 : \mathcal{I} \to \{0,1\}$ of $\mathcal{I}$. In mode $\mu \in [k]$ for some $k \in \mathbb{N}$, two functions $f_i, f_{i'}$ have the same image if $p_\mu(i) = p_\mu(i')$; otherwise $f_i, f_{i'}$ have disjoint images. We conjecture that this generalized notion could be useful in other applications. However, we present separate definitions of both dTFs and 4-to-2 dTFs for the sake of clarity, at the cost of some redundancy.



(a) $\mu = 0$

(b) $\mu = 1$

Figure 3: 4-to-2 Dual-mode trapdoor functions (4-to-2 dTFs): In mode $\mu = 0$, the functions $f_{b_1,b_2}, f_{b_1',b_2'}$ have the same image if and only if $b_1 = b_1'$, and otherwise they have disjoint images. In mode $\mu = 1$, the functions $f_{b_1,b_2}, f_{b_1',b_2'}$ have the same image if and only $b_2 = b_2'$ and otherwise they have disjoint images. The two modes are computationally indistinguishable.

1. ***Efficient key generation.*** *There is an efficient probabilistic algorithm* $\mathsf{Gen}_{\mathcal{F}}$ *which takes as input the security parameter* 1^λ *and a "mode" bit* $\mu \in \{0, 1\}$*, and generates a key* $k(\mu) \in \mathcal{K}$ *together with a trapdoor* t_k*,*

$$(k(\mu), t_k) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu).$$

2. ***Efficient function evaluation.*** *There is an efficient deterministic algorithm that takes as input* $k \in \mathcal{X}$*,* $b_1, b_2 \in \{0, 1\}$*,* $x \in \mathcal{X}$ *and computes* $f_{k,b_1,b_2}(x)$*.*

3. ***Efficient state preparation.*** *There is a quantum polynomial-time algorithm that prepares for every* $k \in \mathcal{K}, b_1, b_2 \in \{0, 1\}$*, a state that is negligibly close in trace distance to the state* $\sum_{x\in\mathcal{X}} \sqrt{D_{k,b_1,b_2}(x)}\,|x\rangle$*.*

4. ***Efficient partial inversion and phase computation with the trapdoor.*** *There exists an efficient deterministic algorithm that takes as input* t_k, y *and outputs the set*

$$\left\{(b_1, b_2) \in \{0, 1\}^2 : \exists x \in \mathcal{X} \text{ s.t. } f_{k,b_1,b_2}(x) = y\right\}.$$

Additionally, there exists another efficient deterministic algorithm that takes as input $y \in \mathcal{Y}$*,* $d \in \bar{\mathcal{X}}$ *and* $b_1, b_2 \in \{0, 1\}$*, along with trapdoor* t_k*, and computes (up to a common normalization factor)* $\alpha_{k,y,d}(b_1, b_2)$*, defined as*

$$\alpha_{k,y,d}(b_1, b_2) := \sum_{x\in\mathcal{X}} (-1)^{d\cdot x} \sqrt{D_{k,b_1,b_2}(x)}\,.$$

When k*,* y *and* d *are clear from the context, we will sometimes abbreviate* $\alpha_{k,y,d}(b_1, b_2)$ *as* α_{b_1,b_2}*.*

5. ***Dual-mode.*** *Intuitively, we want the functions* f_{k,b_1,b_2} *and* $f_{k,b_1',b_2'}$ *to have identical or disjoint images, depending on whether* $b_{\mu+1} = b_{\mu+1}'$*, in mode* μ*. More precisely, for all but a negligible fraction of keys* $(k(\mu), \cdot) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu)$*,*

   (a) *Case 1 (*$b_{\mu+1} \neq b_{\mu+1}'$*): For* $x \leftarrow D_{k,b_1,b_2}$*,* $x' \leftarrow D_{k,b_1',b_2'}$*, we have that* $f_{k,b_1,b_2}(x) \neq f_{k,b_1',b_2'}(x')$ *with probability* 1*.*

(b) *Case 2 ($b_{\mu+1} = b'_{\mu+1}$): For $\tilde{b}_1, \tilde{b}_2 \leftarrow \{0,1\}$, $x \leftarrow D_{k,\tilde{b}_1,\tilde{b}_2}$, $y = f_{k,\tilde{b}_1,\tilde{b}_2}(x)$, for all $d \in \bar{\mathcal{X}}$, with probability at least $1-\varepsilon$, it must be the case that there exists some $s \in \{0,1\}$ such that $\alpha_{b_1,b_2} = (-1)^s \cdot \alpha_{b'_1,b'_2}$.*

6. ***Mode Indistinguishability.*** *Given a key $k(\mu)$ generated with $\mathsf{Gen}(1^\lambda, \mu)$, no efficient algorithm can guess μ with greater than negligible advantage.*

Lemma 4. *Suppose there is an ε-weak dual-mode trapdoor function family $\mathcal{F}$. Then, there exists an ε-weak 4-to-2 dual-mode trapdoor function family $\mathcal{G}$.*

Proof. Suppose $\mathcal{F} = \{f_{k,b} : \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}_\mathcal{F}, b \in \{0,1\}}$ is a ε-weak dual-mode trapdoor function family. Define

$$\mathcal{G} = \{g_{k,b_1,b_2} : \mathcal{X} \times \mathcal{X} \to \mathcal{Y} \times \mathcal{Y}\}_{k \in \mathcal{K}_\mathcal{G}, b_1, b_2 \in \{0,1\}}$$

for some $\mathcal{K}_\mathcal{G} \subseteq \mathcal{K}_\mathcal{F} \times \mathcal{K}_\mathcal{F}$. such that for all $k = (k_1, k_2) \in \mathcal{K}_\mathcal{G}$, $b_1, b_2 \in \{0,1\}$, and $x_1, x_2 \in \mathcal{X}$, define

$$g_{k,b_1,b_2}(x_1, x_2) := \Big(f_{k_1,b_1}(x_1), f_{k_2,b_2}(x_2) \Big).$$

For convenience, we will sometimes write $f_1 = f_{k_1,b_1}$ and $f_2 = f_{k_2,b_2}$. We now show that $\mathcal{G}$ is a 4-to-2 dual-mode trapdoor function family.

Efficient key generation and function evaluation. We define $\mathsf{Gen}_\mathcal{G}(1^\lambda, \mu)$ to work as follows. We sample the key-trapdoor pairs

$$\begin{aligned} (k_1, t_{k_1}) &\leftarrow \mathsf{Gen}_\mathcal{F}(1^\lambda, \mu), \text{ and} \\ (k_2, t_{k_2}) &\leftarrow \mathsf{Gen}_\mathcal{F}(1^\lambda, 1-\mu). \end{aligned}$$

We output the key $k = (k_1, k_2)$ and the trapdoor $t_k = (t_{k_1}, t_{k_2})$. Since $\mathsf{Gen}_\mathcal{F}$ is efficient, so is $\mathsf{Gen}_\mathcal{G}$. Efficient function evaluation follows directly from the definition.

Efficient state preparation. If $\mathcal{D} = \{D_{k,b}\}_{k \in \mathcal{K}_\mathcal{F}, b \in \{0,1\}}$ is the family of probability distributions over $\mathcal{X}$ associated with $\mathcal{F}$. Let $\mathcal{D}' = \{D'_{k,b_1,b_2}\}_{k=(k_1,k_2) \in \mathcal{K}_\mathcal{G}, b_1, b_2 \in \{0,1\}}$ be the family of distributions associated with $\mathcal{G}$, defined as $D'_{k,b_1,b_2}(x_1, x_2) := D_{k_1,b_1}(x_1) \cdot D_{k_2,b_2}(x_2)$. Then we can efficiently prepare the state as a tensor product:

$$\begin{aligned} &\sum_{x_1, x_2 \in \mathcal{X}} \sqrt{D'_{k,b_1,b_2}(x_1, x_2)} \left| x_1, x_2 \right\rangle \\ &\qquad = \sum_{x_1 \in \mathcal{X}} \sqrt{D_{k_1,b_2}(x_1)} \left| x_1 \right\rangle \otimes \sum_{x_2 \in \mathcal{X}} \sqrt{D_{k_2,b_2}(x_2)} \left| x_2 \right\rangle. \end{aligned}$$

Efficient partial inversion with the trapdoor. Given as input $y = (y_1, y_2)$ and the trapdoor $t_k = (t_{k_1}, t_{k_2})$, we can partially invert $f_{k_1,\cdot}$ and $f_{k_2,\cdot}$ on y_1 and y_2 respectively to get $B_1 = \{b_1 : \exists f_{k_1,b_1}(x) = y_1\}$ and $B_2 = \{b_2 : \exists f_{k_2,b_2}(x) = y_2\}$. Output $B_1 \times B_2$.

Efficient phase computation with the trapdoor. Fix $y = (y_1, y_2) \in \mathcal{Y}^2$ and $d = (d_1, d_2) \in \bar{\mathcal{X}}^2$. Then,

$$\begin{aligned}
\alpha_{k,y,d}(b_1, b_2) &= \sum_{x_1, x_2 \in \mathcal{X}} (-1)^{d_1 \cdot x_1 + d_2 \cdot x_2} \cdot \sqrt{D'_{k,b_1,b_2}(x_1, x_2)} \\
&= \sum_{x_1, x_2 \in \mathcal{X}} (-1)^{d_1 \cdot x_1 + d_2 \cdot x_2} \cdot \sqrt{D_{k_1,b_1}(x_1) \cdot D_{k_2,b_2}(x_2)} \\
&= \left(\sum_{x_1 \in \mathcal{X}} (-1)^{d_1 \cdot x_1} \sqrt{D_{k_1,b_1}(x_1)} \right) \cdot \left(\sum_{x_2 \in \mathcal{X}} (-1)^{d_2 \cdot x_2} \sqrt{D_{k_2,b_2}(x_2)} \right) \\
&= \alpha_{k_1,y_1,d_1}(b_1) \cdot \alpha_{k_2,y_2,d_2}(b_2) \ .
\end{aligned} \tag{1}$$

Given the trapdoor, since we can effiently compute $\alpha_{k_1,y_1,d_1}(b_1)$ and $\alpha_{k_2,y_2,d_2}(b_2)$ individually, we can also compute $\alpha_{k,y,d}(b_1, b_2)$.

Dual mode. We will prove the dual-mode property for the case where $\mu = 0$; the case where $\mu = 1$ is symmetric. When $\mu = 0$, the keys $k_1(0)$ and $k_2(1)$ were generated in modes 0 and 1 respectively.

If $b_1 \neq b'_1$, since $k_1(0)$ is in mode 0, we know that the images of f_{k_1,b_1} and f_{k_1,b'_1} are disjoint, and so are the images of g_{k,b_1,b_2} and g_{k,b'_1,b'_2}.

If $b_1 = b'_1$, consider sampling $\tilde{b}_1, \tilde{b}_2 \leftarrow \{0, 1\}$, $x_1 \leftarrow D_{k_1,\tilde{b}_1}$, $x_2 \leftarrow D_{k_2,\tilde{b}_2}$, and setting $y_1 = f_{k_1,\tilde{b}_1}(x_1)$ and $y_2 = f_{k_2,\tilde{b}_2}(x_2)$. Since $k_2(1)$ is in mode 1, we know with probability at least $1 - \varepsilon$, for any $d_2 \in \bar{\mathcal{X}}$ there exists some $s \in \{0, 1\}$ such that $\alpha_{k_2,y_2,d_2}(b_2) = (-1)^s \cdot \alpha_{k_2,y_2,d_2}(b'_2)$. By Equation (1), we get that with probability at least $1 - \varepsilon$ for any $d_1, d_2 \in \bar{\mathcal{X}}$,

$$\begin{aligned}
\alpha_{k,y,d}(b_1, b_2) &= \alpha_{k_1,y_1,d_1}(b_1) \cdot \alpha_{k_2,y_2,d_2}(b_2) = \alpha_{k_1,y_1,d_1}(b'_1) \cdot \alpha_{k_2,y_2,d_2}(b_2) \\
&= \alpha_{k_1,y_1,d_1}(b'_1) \cdot (-1)^s \cdot \alpha_{k_2,y_2,d_2}(b'_2) = (-1)^s \cdot \alpha_{k,y,d}(b'_1, b'_2).
\end{aligned}$$

Mode-Indistinguishability. Consider the following hybrid experiments:

1. Hybrid 1: Generate $k = (k_1, k_2)$ by running $k_1, t_1 \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu)$ and $k_2, t_2 \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, 1 - \mu)$, and output $k = (k_1, k_2)$.

2. Hybrid 2: Generate $k = (k_1, k_2)$ by running $k_1, t_1 \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu)$ and $k_2, t_2 \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu)$, and output $k = (k_1, k_2)$.

3. Hybrid 3: Generate $k = (k_1, k_2)$ by running $k_1, t_1 \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, 1 - \mu)$ and $k_2, t_2 \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda, \mu)$, and output $k = (k_1, k_2)$.

Note that Hybrid 1 corresponds to the setting where k is generated as $k, t \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^\lambda, \mu)$, and Hybrid 3 corresponds to the setting where k is generated as $k, t \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^\lambda, 1-\mu)$. By the mode-indistinguishability of $\mathcal{F}$, Hybrids 1 and 2 are computationally indistinguishable, and similarly Hybrids 2 and 3 are computationally indistinguishable. Therefore, Hybrids 1 and 3 are computationally indistinguishable. This completes the proof. □

3.2 Amplification Lemma for Dual-Mode Trapdoor Functions

In this section, we give an amplification lemma that transforms ε-weak injective and invertible dTFs (see Definition 12) to ε^ℓ-weak dTFs. This transformation is based on the XOR lemma of [SV03].

Lemma 5 (Amplification Lemma for dTFs). *Let λ be the security parameter. Suppose there exists an injective and invertible ε-weak dual-mode trapdoor function family for some $\varepsilon > 0$. Then, there exists a ε^ℓ-weak dual-mode trapdoor function family for any $\ell \leq \mathrm{poly}(\lambda)$.*

Proof. Let $\mathcal{F} = \{f_{k,b} : \mathcal{X} \to \mathcal{Y}\}_{k\in\mathcal{K},b\in\{0,1\}}$ be an injective and invertible, ε-weak dual-mode trapdoor function family. Define the function family $\mathcal{G}$ as

$$\mathcal{G} = \{g_{k,b} : \mathcal{X}^\ell \times \{0,1\}^{\ell-1} \to \mathcal{Y}^\ell\}_{k\in\mathcal{K},b\in\{0,1\}},$$
$$g_{k,b}(x_1,\ldots,x_\ell,r_1,\ldots,r_{\ell-1}) := (y_1,\ldots,y_\ell)\,,$$

where $r_\ell = b \oplus r_1 \oplus \cdots \oplus r_\ell$ and $y_i = f_{k,r_i}(x_i)$ for $i \in [\ell]$. We will show that $\mathcal{G}$ is a ε^ℓ-weak dual-mode trapdoor function family. If $\mathcal{D} = \{D_{k,b}\}_{k,b}$ is the family of probability distributions associated with $\mathcal{F}$, we define a family of probability distributions $\mathcal{D}' = \{D'_{k,b}\}_{k,b}$ associated with $\mathcal{G}$ as follows:

$$D'_{k,b}(x_1,\ldots,x_\ell,r_1,\ldots,r_{\ell-1}) := \frac{1}{2^{\ell-1}} \cdot \prod_{i=1}^{\ell} D_{k,r_i}(x_i),$$

where $r_\ell = b \oplus r_1 \oplus \cdots \oplus r_\ell$.

Key generation, efficient state preparation, and mode indistinguishability. The key generation algorithm for $\mathcal{G}$ is the same as that of $\mathcal{F}$, so the efficient key generation and mode indistinguishability directly for $\mathcal{G}$ follow from the same properties of $\mathcal{F}$. The efficient state preparation property for $\mathcal{G}$ also follows directly from that of $\mathcal{F}$.

Efficient function evaluation. Given the key, the function can also be evaluated efficiently: to evaluate $g_{k,b}$ on input $x_1,\ldots,x_\ell,r_1,\ldots,r_{\ell-1}$, we first compute $r_\ell := b \oplus r_1 \oplus \cdots \oplus r_\ell$. Since $f_{k,\cdot}$ can be efficiently evaluated, we compute $y_i = f_{k,r_i}(x_i)$ for $i \in [\ell]$ and output $(y_1,\ldots,y_\ell)$.

Efficient partial inversion with a trapdoor. There is also a straightforward efficient algorithm that performs partial inversion: given the trapdoor t_k and $y = (y_i)_{i\in[\ell]}$, for each y_i, run the partial inversion algorithm for $\mathcal{F}$ to the set $R_i := \{r \in \{0,1\} : \exists x \text{ s.t. } f_{k,r}(x) = y_i\}$. On a valid input, every R_i must be non-empty; otherwise output $\emptyset$. If $|R_i| = 1$ for every i (so we can write $R_i = \{\hat{r}_i\}$), then there is only on possible value of b, so we output $\{\hat{b}\}$, where $\hat{b} := \hat{r}_1 \oplus \cdots \oplus \hat{r}_\ell$. If for any i, $R_i = \{0,1\}$, this means that y has preimages under both $g_{k,0}$ and $g_{k,1}$, so we output $\{0,1\}$.

For the rest of the proof, we focus on efficient phase computation and the dual-mode property. The proof is by induction, and it will be useful to first set up some notation. For a fixed $y = (y_1,\ldots,y_\ell)$ in the image of $g_{k,b}$, and for a fixed $(d,s) = (d_1,\ldots d_\ell,s_1,\ldots,s_{\ell-1}) \in \bar{\mathcal{X}}^\ell \times \{0,1\}^{\ell-1}$, define $S_{i,b}$ and $\beta_{i,b}$ for each $i \in [\ell]$ as follows:

$$S_{i,b} := \{(x_1,\ldots,x_i,r_1,\ldots,r_{i-1}) : \exists r_i \in \{0,1\} \text{ s.t. } r_1 \oplus \cdots \oplus r_i = b, f_{k,r_1}(x_1) = y_1,\ldots,f_{k,r_i}(x_i) = y_i\}\,,$$

and

$$\beta_{i,b} := \frac{1}{2^{i-1}} \cdot \sum_{\substack{x_1,\ldots,x_i,\\ r_1,\ldots,r_{i-1} \in S_{i,b}}} \left(\prod_{j=1}^{i} (-1)^{d_j \cdot x_j} \cdot \sqrt{D_{k,r_j}(x_j)} \right) \cdot \left(\prod_{j=1}^{i-1} (-1)^{s_j \cdot r_j} \right)$$

$$= \frac{1}{2^{i-1}} \cdot \sum_{\substack{x_1,\ldots,x_i,\\ r_1,\ldots,r_{i-1} \in S_{i,b}}} (-1)^{\sum_{j=1}^{i} d_j \cdot x_j + \sum_{j=1}^{i-1} s_j \cdot r_j} \cdot \left(\prod_{j=1}^{i} \sqrt{D_{k,r_j}(x_j)} \right)$$

Note that $\alpha_{k,y,(d,s)}(b) = \beta_{\ell,b}$, and $S_{\ell,b}$ is the set of all preimages of y under $g_{k,b}$. For $i \in [\ell]$, $r \in \{0,1\}$, let $\mathbb{1}[X_{i,r}]$ be the indicator for the event that there exists some $x_{i,r} \in \mathcal{X}$ such that $f_{k,r}(x_{i,r}) = y_i$. Since $\mathcal{F}$ is an injective and invertible dTF family, such an inverse $x_{i,r}$ must be unique if it exists. Writing $\beta_{i,r}$ in terms of $\beta_{i-1,0}$ and $\beta_{i-1,1}$, we get

$$\beta_{i,b} = \mathbb{1}[X_{i,0}] \cdot \frac{1}{2} \cdot (-1)^{d_i \cdot x_{i,0}} \cdot \sqrt{D_{k,0}(x_{i,0})} \cdot \sum_{\substack{x_1,\ldots,x_{i-1},\\ r_1,\ldots,r_{i-2} \in S_{i-1,b}}} (-1)^{\sum_{j=1}^{i-1} d_j \cdot x_j + \sum_{j=1}^{i-2} s_j \cdot r_j} \cdot \prod_{j=1}^{i-1} \sqrt{D_{k,r_j}(x_j)}$$

$$+ \mathbb{1}[X_{i,1}] \cdot \frac{1}{2} \cdot (-1)^{d_i \cdot x_{i,1} + s_{i-1}} \cdot \sqrt{D_{k,1}(x_{i,1})} \cdot \sum_{\substack{x_1,\ldots,x_i\\ r_1,\ldots,r_{i-2}\\ \in S_{i-1,1-b}}} (-1)^{\sum_{j=1}^{i-1} d_j \cdot x_j + \sum_{j=1}^{i-2} s_j \cdot r_j} \cdot \prod_{j=1}^{i-1} \sqrt{D_{k,r_j}(x_j)}$$

$$= \frac{1}{2} \sqrt{D_{k,0}(x_{i,0})} \cdot \left(\mathbb{1}[X_{i,0}] \cdot (-1)^{d_i \cdot x_{i,0}} \cdot \beta_{i-1,b} + \mathbb{1}[X_{i,1}] \cdot (-1)^{d_i \cdot x_{i,1} + s_{i-1}} \cdot \beta_{i-1,1-b} \right) , \tag{2}$$

where the last inequality follows since $D_{k,0}(x_{i,0}) = D_{k,1}(x_{i,1})$, since $\mathcal{F}$ is an injective and invertible dTF.

Efficient Phase Computation with Trapdoor. We observe that in the base case,

$$S_{1,b} = \{x_1 : f_{k,b}(x_1) = y_1\},$$

and since $\mathcal{F}$ is an injective and invertible dTF family, $\beta_{1,b}$ is efficiently computable. For the same reason, the phases $(-1)^{d_i \cdot x_{i,0}}$ and $(-1)^{d_i \cdot x_{i,1} + s_{i-1}}$ are efficiently computable for all $i \in [\ell]$. Then, by Equation (2), we can sequentially compute $\beta_{i,r}$ for $i = 1, \ldots, \ell$, and $r \in \{0,1\}$, to get finally $\alpha_b = \alpha_{k,y,(d,s)}(b) = \beta_{\ell,b}$.

Dual-mode. In the disjoint mode ($\mu = 0$), since the images of $f_{k,0}$ and $f_{k,1}$ are disjoint, for each $i \in [\ell]$, there exist unique r_i, x_i such that $f_{k,r_i}(x_i) = y_i$. This implies that there is a unique $b = r_1 \oplus \cdots \oplus r_\ell$ such that y is in the image of $g_{k,b}$. In the lossy mode ($\mu = 1$), for $i \in [\ell]$ letting $z_i \in \{0,1\}$ such that $(-1)^{z_i} = (-1)^{d_i \cdot x_{i,1} + s_{i-1} - d_i \cdot x_{i,0}}$, Equation (2) implies

$$|\beta_{i,b}| = \frac{1}{2} \cdot \sqrt{D_{k,0}(x_{i,0})} \cdot \left| \mathbb{1}[X_{i,0}] \cdot \beta_{i-1,b} + \mathbb{1}[X_{i,1}] \cdot (-1)^{z_i} \cdot \beta_{i-1,1-b} \right| .$$

If $\mathbb{1}[X_{i,0}] = 1$ and $\mathbb{1}[X_{i,1}] = 0$, we have that $|\beta_{i,b}| = \frac{1}{2} \cdot \sqrt{D_{k,0}(x_{i,0})} \cdot |\beta_{i-1,b}|$. Symmetrically, if on the other hand $\mathbb{1}[X_{i,0}] = 0$ and $\mathbb{1}[X_{i,1}] = 1$, we have that $|\beta_{i,b}| = |\beta_{i-1,1-b}|$. In the case where $\mathbb{1}[X_{i,0}] = \mathbb{1}[X_{i,1}] = 1$,

$$|\beta_{i,b}| = \frac{1}{2} \cdot \sqrt{D_{k,0}(x_{i,0})} \cdot \left| \beta_{i-1,b} + (-1)^{z_i} \beta_{i-1,1-b} \right| = |\beta_{i,1-b}| .$$

Therefore, in all cases, if $|\beta_{i-1,0}| = |\beta_{i-1,1}|$, we have that $|\beta_{i,0}| = |\beta_{i,1}|$.

All that remains is to prove the base case: we want to show that there exists some $i^* \in [\ell]$ for which $|\beta_{i^*,0}| = |\beta_{i^*,1}|$. By the lossy-mode property of ε-weak dTF $\mathcal{F}$, with probability[6] at least $1-\varepsilon^{\ell}$, there exists some $i \in [\ell]$ such that $\mathbb{1}[X_{i,0}] = \mathbb{1}[X_{i,1}] = 1$. Let i^* be the smallest such i. For all $j < i^*$, exactly one of $|\beta_{j,0}|$ and $|\beta_{j,1}|$ is nonzero, the other being 0. Further, $|\beta_{i^*,0}| = |\beta_{i^*,1}| = \frac{1}{2} \cdot \sqrt{D_{k,0}(x_{i^*,0})}$, satisfying the base case for our induction. Therefore in the lossy mode, with probability at least $1-\varepsilon^{\ell}$, $|\beta_{\ell,0}| = |\beta_{\ell,1}|$. Since the $\beta_{i,r}$'s are real-valued, this means that there exists some $s \in \{0,1\}$ such that $\beta_{\ell,0} = (-1)^s \cdot \beta_{\ell,1}$. This completes the proof. □

4 Remote State Preparation of the DSS Gadget

We construct our QFHE scheme by dequantizing the client in the scheme of Dulek, Schaffner and Speelman [DSS16], which we will henceforth refer to as DSS. The DSS QFHE scheme builds on the Pauli one-time pad-based template of Broadbent and Jeffery [BJ15]. Concretely, a quantum state $|\psi\rangle$ is encrypted by applying a random Pauli one-time pad to get $|\tilde{\psi}\rangle := \mathsf{X}^x\mathsf{Z}^z |\psi\rangle$. The keys x, z of the one-time pad are encrypted under a classical homomorphic encryption scheme to get ciphertexts $\tilde{x}, \tilde{z}$. The one-time padded state $|\tilde{\psi}\rangle$ along with these encryptions $\tilde{x}, \tilde{z}$ form the encryption of the state $|\psi\rangle$.

Evaluating any Clifford gate is easy, since Clifford gates nicely "commute" through the Pauli one-time pad; we only need to update the one-time pads homomorphically according to standard rules of Clifford-Pauli commutation. On the other hand when the non-Clifford T gate is applied to a Pauli one-time padded state, we end up with an extra non-Pauli error in the form of a P gate, since

$$\mathsf{T}\mathsf{X}^x\mathsf{Z}^z |\psi\rangle = \mathsf{P}^x\mathsf{X}^x\mathsf{Z}^z\mathsf{T} |\psi\rangle .$$

The natural solution would be to apply a $\mathsf{P}^\dagger$ gate to this state if $x = 1$, and otherwise do nothing. However it is not clear how to do this, because the evaluator only has access to an encryption $\tilde{x}$ of the one-time pad key x.

Dulek et al. [DSS16] solve this issue by "obfuscating" the decryption algorithm (hard-coded with the secret key sk) in the form of a quantum teleportation gadget $\Gamma(sk)$, henceforth called the DSS gadget or the DSS teleportation gadget. The client prepares this (quantum) gadget during key generation, and sends it to the evaluator as part of the evaluation key. When the qubit $\mathsf{P}^x\mathsf{X}^x\mathsf{Z}^z\mathsf{T} |\psi\rangle$ is teleported through the gadget (in a way that depends on $\tilde{x}$), the gadget implicitly decrypts $\tilde{x}$ and applies the $(\mathsf{P}^\dagger)^x$ correction to the teleported qubit. More specifically, we can think of running the DSS gadget as a blackbox that takes as input a state ρ and an encryption $\mathsf{HE.Enc}_{pk}(x)$ for some $x \in \{0,1\}$ and outputs the state

$$\mathsf{X}^{x'}\mathsf{Z}^{z'}(\mathsf{P}^\dagger)^x\rho(\mathsf{P})^x\mathsf{Z}^{z'}\mathsf{X}^{x'},$$

for some one-time pad keys $x', z' \in \{0,1\}$, along with encryptions $\mathsf{HE.Enc}_{pk'}(x')$ and $\mathsf{HE.Enc}_{pk'}(z')$ under an independent public key pk'. This functionality is visualized in Figure 1, up to a renaming of the variables x', z'.

In this section, we describe a protocol by which the evaluator is able to prepare the DSS gadget $\Gamma(sk)$ on its own, with the help of some *classical* information generated by the client during the key generation procedure. In other words, we give a remote state preparation protocol for the DSS gadget. As a result, the client is entirely classical. This protocol is described in Section 4.3. The complete description of the

[6] the probability is taken over the choice of y, which is sampled by first sampling $x_1, \ldots, x_\ell \leftarrow \mathcal{X}$ and $r_1, \ldots, r_{\ell-1} \leftarrow \{0,1\}$ and setting $y = g_{k,b}(x_1, \ldots, x_\ell, r_1, \ldots, r_{\ell-1})$. Let i^* be the smallest such i.

structure of the DSS gadget is quite involved, so we give only an abstract description of its structure in Section 4.1, which will suffice for our purposes. We defer the details of the structure of the DSS gadget to Appendix A.

At the heart of our remote state preparation protocol is a simpler protocol that takes as input a three-qubit state, say on registers labeled $0, 1, 2$, along with a "4-to-2" dual-mode trapdoor function key $k(\mu)$ generated in mode μ. The output is the (one-time padded) state $|\Phi^+\rangle_{2,\mu} \otimes |+\rangle_{1-\mu}$, where the qubit in register μ forms a Bell pair with qubit 2, and these two registers are in tensor product with the qubit in register $(1-\mu)$. In other words, this protocol allows the remote state preparation of a Bell pair between two of three qubits, while hiding the identity of the qubits that form the Bell pair. This protocol for remote state preparation of a "hidden Bell pair" is described in Section 4.2.

4.1 An abstract description of the DSS gadget

The DSS gadget is composed of a set of entangled qubits to form a perfect matching of Bell pairs. To facilitate teleportation, the gadget also defines a set of Bell-basis measurements for the evaluator to perform.

The complete description of this matching and measurement instructions is quite involved, so we defer our the exposition to Appendix A. For the purpose of this section, we only need an abstract specification of the structure of the DSS gadget, which we will now describe.

The DSS gadget is defined by a tuple $(Q, Q_1, Q_2, P, \pi, \nu)$, where each element of the tuple is described below. Let $n = |sk|$ be the length of the secret key, and let $\mathcal{C}$ be the space of ciphertexts for a single bit message for the classical homomorphic encryption scheme HE.

- Q is the set of qubit registers in the DSS gadget.
- (Q_1, Q_2) is a partition of Q, across which the entangled Bell pairs form a bipartite graph.
- $P \subseteq Q$ is the set of registers with an inverse phase gate $\mathsf{P}^\dagger$ applied.
- $\pi : \{0,1\}^n \times Q_1 \to Q_2$ is an injective function that defines the matching of qubits that form the maximally entangled Bell pairs of the gadget. If sk is the secret key, then for every $q \in Q_1$, the qubits in registers $q, \pi(sk, q)$ form a Bell pair. We write π^{sk} as shorthand for $\pi(sk, \cdot)$.[7]
- $\nu : Q_2 \times \mathcal{C} \to Q_1 \cup \{\perp\}$ defines the measurements the evaluator must perform to teleport the input qubit through the gadget. These measurements depend on the ciphertext $\tilde{x}$, which the evaluator has access to. For every $q \in Q_2$, the evaluator performs a Bell basis measurement on the qubits $q, \nu(\tilde{x}, q)$ (in a specified order) and records the measurements to update the one-time pad keys.

All of Q, Q_1, Q_2, P, π, ν are public and known to the evaluator. Together with the secret key sk, they determine the structure of the DSS gadget, that is, they determine which qubit pairs are entangled to form Bell pairs. The function $\pi^{sk} := \pi(sk, \cdot)$ defines this bijection from Q_1 to Q_2. Concretely, the gadget $\Gamma(sk)$ consists of the state

$$\gamma_{x,z}(sk) := \bigotimes_{q \in Q_1} \mathsf{X}^{x_q} \mathsf{Z}^{z_q} (\mathsf{P}^\dagger)^{p_q} \left|\Phi^+\right\rangle \left\langle \Phi^+\right|_{q, \pi^{sk}(q)} (\mathsf{P})^{p_q} \mathsf{Z}^{z_q} \mathsf{X}^{x_q},$$

where $x, z \in \{0,1\}^{|Q_1|}$ are one-time pad keys and $p \in \{0,1\}^{|Q_1|}$ is defined as $p_q = 1$ if $q \in P$ and 0 otherwise. The gadget $\Gamma(sk)$ also includes classical encryptions $\tilde{x}, \tilde{z}$ of the one-time pad keys under an independent public key pk'. We write $\Gamma_{pk'}(sk) = (\gamma_{x,z}(sk), \tilde{x}, \tilde{z})$ to denote this.

[7] Although π is public and known to the evaluator, π^{sk} is still private and known only to the client.

The function π has the additional property that for any $q \in Q_1$, there exists some $i(q)$ such that $sk_{i(q)}$ determines the value of $\pi(\cdot, q)$. That is, $\pi(\cdot, q)$ can take one of two distinct values, and depends on only the $i(q)$th bit of sk (as opposed to some global function of sk). We denote these two possibilities as $\pi_0(q)$ and $\pi_1(q)$, the value of $\pi(sk, q)$ when $sk_{i(q)} = 0$ and $sk_{i(q)} = 1$ respectively.

4.2 Remotely Preparing a Hidden Bell Pair

In this subsection, we give a protocol to remotely prepare a hidden Bell pair. At a high level, given a key $k(\mu)$ of a 4-to-2 dual-mode trapdoor function (which can be instantiated given just a dual-mode trapdoor function), and three registers in the state $|+\rangle_0 \otimes |+\rangle_1 \otimes |+\rangle_2$, the protocol produces a one-time padded version of the following state

$$\left|\Phi^+\right\rangle_{2,\mu} \otimes |+\rangle_{1-\mu} .$$

To use this protocol in the remote state preparation of the DSS gadget, we also need to consider input states which are one-time padded with Z gates and the states in registers 0 and 1 could be potentially entangled with other registers.

Lemma 6. *Let $\mathcal{F}$ be a ε-weak 4-to-2 dual-mode trapdoor function family, and let HE be a fully homomorphic encryption scheme. For a bit $\mu \in \{0,1\}$, let $k, t \leftarrow \mathsf{KeyGen}_{\mathcal{F}}(1^\lambda, \mu)$ be a key-trapdoor pair, let $(pk, evk, sk) \leftarrow \mathsf{HE.KeyGen}(1^\lambda)$ be a key set, and let $\tilde{t} \leftarrow \mathsf{HE.Enc}_{pk}(t)$ be an encryption of the trapdoor t under pk. Consider a BQP machine with access to the state*[8]

$$(\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{I})\Big(|\psi_0\rangle_0 \otimes |\psi_1\rangle_1 \otimes |+\rangle_2\Big) \tag{3}$$

on three registers (labelled $0, 1, 2$), where $|\psi_i\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \left|\phi_i^0\right\rangle + |1\rangle \left|\phi_i^1\right\rangle\right)$ for $i \in \{0,1\}$. Let the BQP machine also have access to $k, \tilde{t}, pk, evk$ as well as encryptions of z_0, z_1 under pk. Then, there is a BQP algorithm that computes a state that is within negligible trace distance from the state

$$\frac{1}{\sqrt{2}}(\mathsf{Z}^{r_0} \otimes \mathsf{Z}^{r_1} \otimes \mathsf{X}^{r_2})\Big((|00\rangle \left|\phi_\mu^0\right\rangle + |11\rangle \left|\phi_\mu^1\right\rangle)_{2,\mu} \otimes |\psi_{1-\mu}\rangle_{1-\mu}\Big), \tag{4}$$

along with encryptions of r_0, r_1, r_2 under pk, with probability at least $1 - \varepsilon$.

Proof. First, we will introduce some notation that will be useful throughout the proof. Define the function $i : \{0,1\}^3 \to \{0,1\}^2$ to be $i(u,v,w) := (u \oplus v, u \oplus w)$. Note that $i(u,v,w) = i(1-u, 1-v, 1-w)$. For readability, we will write $f_{i(u,v,w)}$ or f_{uvw} to denote $f_{k,i(u,v,w)}$, depending on the context, where k is the key given as an input. We will also abbreviate $D_{k,i(u,v,w)}$ as $D_{i(u,v,w)}$ or D_{uvw}.

We can rewrite the input state as

$$\left|\chi^{(1)}\right\rangle = (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{I})\,\frac{1}{2\sqrt{2}} \sum_{u,v,w \in \{0,1\}} (|v\rangle \left|\phi_0^v\right\rangle)_0 \otimes (|w\rangle \left|\phi_1^w\right\rangle)_1 \otimes |u\rangle_2 \ .$$

By controlling appropriately on u, v, w, we can then prepare a state that is negligibly close in trace distance to the following state:

$$\begin{aligned}\left|\chi^{(2)}\right\rangle &= (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{I} \otimes \mathsf{I}) \\ &\sum_{\substack{u,v,w \in \{0,1\} \\ x \in \mathcal{X}}} \frac{1}{2\sqrt{2}} \cdot \sqrt{D_{uvw}(x)} \cdot (|v\rangle \left|\phi_0^v\right\rangle)_0 \otimes (|w\rangle \left|\phi_1^w\right\rangle)_1 \otimes |u\rangle_2 \otimes |x\rangle_3 \,.\end{aligned}$$

[8] The single-qubit gates $\mathsf{Z}^{z_0}, \mathsf{Z}^{z_1}$ are applied to the first qubits of the 0, 1 registers.

We now use the evaluation algorithm for $\mathcal{F}$ to coherently evaluate f_{uvw} in register 4.

$$\left|\chi^{(3)}\right\rangle = (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{I} \otimes \mathsf{I} \otimes \mathsf{I})$$
$$\sum_{\substack{u,v,w \in \{0,1\} \\ x \in \mathcal{X}}} \frac{\sqrt{D_{uvw}(x)}}{2\sqrt{2}} \cdot (|v\rangle\, |\phi_0^v\rangle)_0 \otimes (|w\rangle\, |\phi_1^w\rangle)_1 \otimes |u\rangle_2 \otimes |x\rangle_3 \otimes |f_{uvw}(x)\rangle_4 \,.$$

Now, we measure the fourth register to obtain some $y \in \mathcal{Y}$. At this point it is useful to introduce some more notation: let $\mathcal{X}_{b_1,b_2}$ be the set of all pre-images of y under f_{b_1,b_2}, that is, $\mathcal{X}_{b_1,b_2} = \{x \mid y = f_{b_1,b_2}(x)\}$. Again, we will abuse notation and write $\mathcal{X}_{uvw}$ to denote $\mathcal{X}_{i(u,v,w)}$. Finally, let $\mathcal{Y}_{b_1,b_2}$ the image of f_{b_1,b_2} when evaluated on the support of D_{b_1,b_2}.

For simplicity, let us first consider the case where $\mu = 0$. The case where $\mu = 1$ is symmetric. Let $a \in \{0,1\}$ be such that $y \in \mathcal{Y}_{a,0} \cup \mathcal{Y}_{a,1}$. By the dual-mode property we know that there is exactly one such a. (Note that a is not known to the evaluator in the clear at this point.) After the measurement of the fourth register, the superposition on the remaining registers collapses to have only values of u, v, w and x such that $y = f_{uvw}(x)$. In this case, since $y \in \mathcal{Y}_{a,0} \cup \mathcal{Y}_{a,1}$, we must have that $u \oplus v = a$. The resulting state is (up to normalization factors)

$$\begin{aligned}\left|\chi^{(4)}\right\rangle &\propto (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{I} \otimes \mathsf{I}) \cdot \\ &\sum_{\substack{u,v,w \in \{0,1\} \\ u \oplus v = a \\ x \in \mathcal{X}_{uvw}}} \sqrt{D_{uvw}(x)} \cdot (|v\rangle\, |\phi_0^v\rangle)_0 \otimes (|w\rangle\, |\phi_1^w\rangle)_1 \otimes |u\rangle_2 \otimes |x\rangle_3 \\ &= (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{I} \otimes \mathsf{I}) \cdot \\ &\sum_{\substack{v,w \in \{0,1\} \\ x \in \mathcal{X}_{a,a\oplus v\oplus w}}} \sqrt{D_{a,a\oplus v\oplus w}(x)} \cdot (|v\rangle\, |\phi_0^v\rangle)_0 \otimes (|w\rangle\, |\phi_1^w\rangle)_1 \otimes |v \oplus a\rangle_2 \otimes |x\rangle_3 \,.\end{aligned}$$

Rearranging the registers and writing $|v \oplus a\rangle$ as $\mathsf{X}^a\, |v\rangle$, we see that we almost have the desired state as in Equation 4, except we want to uncompute register 3,

$$\begin{aligned}\left|\chi^{(4)}\right\rangle \propto &(\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{X}^a \otimes \mathsf{I}) \cdot \\ &\sum_{\substack{v,w \in \{0,1\} \\ x \in \mathcal{X}_{a,a\oplus v\oplus w}}} \sqrt{D_{a,a\oplus v\oplus w}(x)} \cdot (|v\rangle \otimes |v\rangle\, |\phi_0^v\rangle)_{2,0} \otimes (|w\rangle\, |\phi_1^w\rangle)_1 \otimes |x\rangle_3 \,.\end{aligned}$$

Towards this end, we measure register 3 in the Hadarmard basis. First applying the Hadamard transform on register 3, we get a state proportional to

$$(\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{X}^a \otimes \mathsf{I}) \cdot$$
$$\sum_{\substack{v,w \in \{0,1\} \\ x \in \mathcal{X}_{a,a\oplus v\oplus w}}} \sum_{d \in \bar{\mathcal{X}}} (-1)^{d \cdot x} \cdot \sqrt{D_{a,a\oplus v\oplus w}(x)} \cdot (|v\rangle \otimes |v\rangle\, |\phi_0^v\rangle)_{2,0} \otimes (|w\rangle\, |\phi_1^w\rangle)_1 \otimes |d\rangle_3 \;.$$

Measuring register 3, suppose we get outcome $d \in \bar{\mathcal{X}}$. The resulting state is then proportional to

$$\left|\chi^{(5)}\right\rangle \propto (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{X}^{a} \otimes \mathsf{I}) \sum_{v,w \in \{0,1\}} \sum_{x \in \mathcal{X}_{a,a\oplus v\oplus w}} (-1)^{d\cdot x} \cdot \sqrt{D_{a,a\oplus v\oplus w}(x)} \cdot (|v\rangle \otimes |v\rangle\, |\phi_0^v\rangle)_{2,0} \otimes (|w\rangle\, |\phi_1^w\rangle)_1 .$$

For convenience, let

$$\beta_\nu := \alpha_{k,y,d}(a, a\oplus \nu) = \sum_{x \in \mathcal{X}_{a,a\oplus\nu}} (-1)^{d\cdot x} \cdot \sqrt{D_{a,a\oplus\nu}(x)}\ ,$$

so that we can rewrite the state as

$$\left|\chi^{(5)}\right\rangle \propto (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{X}^{a} \otimes \mathsf{I}) \sum_{v,w \in \{0,1\}} \beta_{v\oplus w} \cdot (|v\rangle \otimes |v\rangle\, |\phi_0^v\rangle)_{2,0} \otimes (|w\rangle\, |\phi_1^w\rangle)_1 .$$

We know that with probability at least $1-\varepsilon$ over the measurement outcome y, $\beta_0 \in \{\pm\beta_1\}$. Let $s \in \{0,1\}$ be such that $\beta_1 = (-1)^s \cdot \beta_0$, we can factor out a global phase of β_0 and equivalently write the state as follows:

$$\begin{aligned}\left|\chi^{(5)}\right\rangle &\propto (\mathsf{Z}^{z_0} \otimes \mathsf{Z}^{z_1} \otimes \mathsf{X}^{a} \otimes \mathsf{I}) \sum_{v,w \in \{0,1\}} (-1)^{s\cdot(v\oplus w)} (|v\rangle \otimes |v\rangle\, |\phi_0^v\rangle)_{2,0} \otimes (|w\rangle\, |\phi_1^w\rangle)_1 \\ &= (\mathsf{Z}^{z_0\oplus s} \otimes \mathsf{Z}^{z_1\oplus s} \otimes \mathsf{X}^{a} \otimes \mathsf{I}) \sum_{v,w \in \{0,1\}} (|v\rangle \otimes |v\rangle\, |\phi_0^v\rangle)_{2,0} \otimes (|w\rangle\, |\phi_1^w\rangle)_1 .\end{aligned}$$

We can now homomorphically compute encryptions of the new one-time pad keys, since we are also given an encryption of the trapdoor. First, by the property of of efficient partial inversion, we can obtain all possible values of (b_1, b_2) such that $y \in \mathcal{Y}_{b_1,b_2}$, and therefore obtain (an encryption of) the value of $r_2 = a$. Next, by the property of efficient phase computation, we can homomorphically compute β_0, β_1 and therefore s, which allows us to compute encryptions of $r_0 = z_0 \oplus s$ and $r_1 = z_1 \oplus s$. □

4.3 Remotely Preparing the DSS Gadget

We dequantize the DSS client by giving a protocol $\mathsf{GadgetKeyGen}(sk, pk')$ that allows the key generation to be classical, so that it outputs classical "gadget keys" $gk(sk)$ that depend on but computationally hide a secret key sk. Using the gadget keys gk, the evaluator can then run $\mathsf{RSPGenGadget}$ to prepare a teleportation gadget state $\Gamma_{pk'}(sk)$ that can then be used as in [DSS16] for the evaluation of T gates. The gadget key must also include trapdoor information encrypted under an independent public key pk', so that the evaluator can homomorphically update its one-time pad key encryptions.

The client's procedure for generating the gadget keys.

Suppose the length of the secret key is $|sk| = n$ bits. The client uses this procedure to "encrypt" the secret key sk (in the form of keys of a "4-to-2" dual-mode trapdoor function family $\mathcal{F}$), which provide instructions to the evaluator to prepare teleportation gadget $\Gamma_{pk'}(sk)$. The procedure is as follows.

$\underline{\mathsf{GadgetKeyGen}(sk, pk')}$

1. For every $i \in [n]$, generate key-trapdoor pairs $k_i, t_i \leftarrow \mathsf{KeyGen}_{\mathcal{F}}(1^\lambda, sk_i)$, giving the ith bit of sk as input.
2. Encrypt the trapdoors t_i using the public key pk', to get ciphertexts $\tilde{t}_i \leftarrow \mathsf{HE.Enc}_{pk'}(t_i)$ for all $i \in [n]$.
3. Output the gadget key $gk = (k_i, \tilde{t}_i)_{i=1}^n$.

The evaluator's procedure for preparing the DSS gadget.

The evaluator uses the gadget keys generated by the client to prepare the DSS gadget. The procedure works as follows.

$\underline{\mathsf{RSPGenGadget}(gk, \tilde{sk}, pk', evk')}$

1. Parse the gadget key as $gk = (k_i, \tilde{t}_i)_{i=1}^n$.
2. Prepare $|Q|$ qubit registers, each holding state $|+\rangle$.
3. Maintain encryptions $\tilde{x}_q, \tilde{z}_q$ of the one-time pad keys for each register $q \in Q_1$ under pk'. Initialize all of these to encryptions of 0.
4. For every $q \in Q_1$, run the protocol described in Lemma 6 on registers $\pi_0(q)$, $\pi_1(q)$ and q in the place of registers labeled 0, 1 and 2 in the order. Let $i(q) \in [n]$ be such that the $i(q)$th bit $sk_{i(q)}$ of sk determines the value of $\pi^{sk}(q)$ Also as input to the protocol to prepare the hidden Bell pair, provide $k_{i(q)}, \tilde{t}_{i(q)}, \tilde{sk}, pk', evk'$ and $\tilde{z}_{\pi_0(q)}$ and $\tilde{z}_{\pi_1(q)}$.
5. As the output of the hidden Bell pair procedure, we get encryptions of the updated one-time pad keys, so we replace $\tilde{x}_q$, $\tilde{z}_{\pi_0(q)}$ and $\tilde{z}_{\pi_1(q)}$ with the updated keys.
6. Finally, apply an inverse phase gate $\mathsf{P}^\dagger$ to the qubits in registers $q \in P$. Since $\mathsf{P}^\dagger$ is Clifford, simply update the encrypted one-time pad keys to reflect this change.

Lemma 7. *Given a gadget key $gk \leftarrow$ $\mathsf{GadgetKeyGen}(sk, pk')$, public key pk' and evaluation key evk',* $\mathsf{RSPGenGadget}$ *correctly outputs $\Gamma_{pk'}(sk) = (\gamma_{x,z}(sk), \tilde{x}, \tilde{z})$, where*

$$\gamma_{x,z}(sk) := \bigotimes_{q \in Q_1} \mathsf{X}^{x_q}\mathsf{Z}^{z_q}(\mathsf{P}^\dagger)^{p_q} \left|\Phi^+\right\rangle \left\langle\Phi^+\right|_{q,\pi^{sk}(q)} (\mathsf{P})^{p_q}\mathsf{Z}^{z_q}\mathsf{X}^{x_q}, \tag{5}$$

where $x, z \in \{0,1\}^{|Q_1|}$ are one-time pad keys and $\tilde{x}$ and $\tilde{z}$ are their encryptions under public key pk'.

Proof. We argue that throughout the $\mathsf{RSPGenGadget}(gk, pk', evk')$, the inputs to the procedure in Lemma 6, to prepare hidden Bell pairs, are of the form of Equation (3). As $\mathsf{RSPGenGadget}$ processes the registers $q \in Q_1$ one at a time, it prepares the corresponding component $|\Phi^+\rangle \langle\Phi^+|_{q,\pi^{sk}(q)}$ of the tensor product state $\gamma_{x,z}(sk)$. In the beginning, for the first register $q \in Q_1$, since all the registers are in the state $|+\rangle$, the input condition is satisfied, with $|\psi_0\rangle = |\psi_1\rangle = |+\rangle$. Let $\mu = sk_{i(q)}$, where $i(q) \in [n]$ is such that $sk_{i(q)}$ determines the value of $\pi^{sk}(q)$. After running the hidden Bell pair procedure, we obtain the state

$$(\mathsf{Z}^{r_0} \otimes \mathsf{Z}^{r_1} \otimes \mathsf{X}^{r_2}) \Big((|00\rangle + |11\rangle)_{2,\mu} \otimes |+\rangle_{1-\mu} \Big),$$

as desired. For every subsequent $q \in Q_1$, the register q is untouched thus far and therefore holds the $|+\rangle$ state. If $\mu = sk_{i(q)}$ and $q' = \pi^{sk}(q) \in Q_2$, we know that q' cannot be already entangled to another qubit in Q_1 since π^{sk} is injective. Therefore the state on register μ is $\mathsf{Z}^{z_\mu} |\psi_\mu\rangle$, where $|\psi_\mu\rangle = |+\rangle$. Further, the state on register $1 - \mu$ must be in one of the following two states: either it was previously untouched, in which case it simply holds the state $|+\rangle$, or it is entangled with some other $\bar{q} \in Q_1$, in which case their joint state is of the form

$$(\mathsf{Z}^{z_{1-\mu}} \otimes \mathsf{X}^r) |\psi_{1-\mu}\rangle = (\mathsf{Z}^{z_{1-\mu}} \otimes \mathsf{X}^r) (|00\rangle + |11\rangle)_{1-\mu,\bar{q}} \ ,$$

for some $z_{1-\mu}, r \in \{0, 1\}$. Again, this satisfies the input conditions in Equation (3). Therefore the procedure in Lemma 6 succeeds in correctly preparing the DSS gadget state (up to negligible trace distance error). □

5 Our QFHE scheme

In this section, we give a quantum (leveled) fully homomorphic encryption scheme, built in a black-box way from compact classical fully homomorphic encryption and dual-mode trapdoor functions.

Theorem 2. *Let λ be the security parameter. Suppose there exists a dual-mode trapdoor function family $\mathcal{F}$. Suppose further that there exists a leveled fully homomorphic encryption scheme* HE *with a decryption algorithm* HE.Dec *in* NC_1*, that is,* HE.Dec *is computable by a boolean function fan-in* 2 *circuit of depth $O(\log(\lambda))$. Then, there exists a compact quantum leveled fully homomorphic encryption scheme* QFHE *with a classical key generation procedure.*

We will now describe our QFHE scheme.

Key Generation. Let L be the maximum T-depth of the circuits we want to homomorphically evaluate. Then, the client generates the keys as follows.

$\underline{\mathsf{QFHE.KeyGen}(1^\lambda, 1^L)}$

1. For $i = 0$ to L, independently generate key $(pk_i, sk_i, evk_i) \leftarrow \mathsf{HE.KeyGen}(1^\lambda)$.
2. Set the public key as $pk = (pk_i)_{i=0}^L$.
3. Set the secret key as $sk = (sk_i)_{i=0}^L$.
4. For $i = 0$ to $L-1$, compute $gk_i \leftarrow \mathsf{GadgetKeyGen}(sk_i, pk_{i+1})$. Compute an encryption of sk_i under pk_{i+1}. Set the evaluation key evk to be

$$\left(gk_i, \mathsf{HE.Enc}_{pk_{i+1}}(sk_i)\right)_{i=0}^{L-1}, (evk_i)_{i=0}^L$$

Encryption. The encryption procedure is identical to that of [DSS16]. For completeness, we descibe it here.

$\underline{\mathsf{QFHE.Enc}_{pk}(|\psi\rangle)}$

1. Let $|\psi\rangle$ be an n-qubit input state. Sample uniformly random keys $x, z \leftarrow \{0,1\}^n$. Parse the public key as $pk = (pk_i)_{i=0}^{L}$.
2. Output the quantum one-time-padded state $\mathsf{X}^x\mathsf{Z}^z\,|\psi\rangle$ along with encryptions of x, z under the first public key pk_0.

Note that if the input is classical, $|\psi\rangle$ is just a standard basis state and the encryption is also classical.

Circuit Evaluation. The circuit evaluation procedure is almost the same as that of [DSS16], except that the evaluator needs to construct the teleportation gadget each time for each T gate it needs to perform. For completeness, we provide a description of the entire circuit evaluation procedure as in [DSS16] with this modification. Without loss of generality, the quantum circuit C we want to evaluate is written using a universal gate set consisting of the Clifford group generators $\{\mathsf{H}, \mathsf{P}, \mathsf{CNOT}\}$ and the (non-Clifford) T gate. Note that while the Clifford gates might affect multiple qubits at the same time, the T gates only affect a single qubit. Before evaluation of a gate U, the encryption of the n-qubit state ρ is of the form

$$X^x Z^z \rho X^x Z^z = (\mathsf{X}^{x_1}\mathsf{Z}^{z_1} \otimes \cdots \otimes \mathsf{X}^{x_n}\mathsf{Z}^{z_n})\rho(\mathsf{X}^{x_1}\mathsf{Z}^{z_1} \otimes \cdots \otimes \mathsf{X}^{x_n}\mathsf{Z}^{z_n})^\dagger.$$

The evaluating party also holds encryptions $\tilde{x}_1^{[i]}, \ldots, \tilde{x}_n^{[i]}$ and $\tilde{z}_1^{[i]}, \ldots, \tilde{z}_n^{[i]}$ of the one-time pad keys, with respect to the ith key set pk_i for some i (initially, $i = 0$). The goal is to produce an encrypted state $(\mathsf{X}^{x'}\mathsf{Z}^{z'})U\rho U^\dagger(\mathsf{X}^{x'}\mathsf{Z}^{z'})^\dagger$, along with encryptions of the new keys x', z' of the quantum one-time pad. If U is a Clifford gate, these encryptions will remain in the ith key set, but if U is a T gate, all the encryptions are transferred to the $(i+1)$the key set.

$\underline{\mathsf{QFHE.Eval}(C, \rho, \tilde{x}^{[0]}, \tilde{z}^{[0]}, evk)}$

- If U is a Clifford gate, the evaluator simply applies U to the encrypted qubit. Since U commutes with the Pauli group, the evaluator only needs to update the encrypted keys in a straightforward way.
- If $U = \mathsf{T}$, the evaluator applies the T gate to the appropriate wire w. Supposing the qubit at wire w starts off in the state $(\mathsf{X}^{x_w}\mathsf{Z}^{z_w})\rho_w(\mathsf{X}^{x_w}\mathsf{Z}^{z_w})^\dagger$, after the T gate is applied, it is now in the state

  $$(\mathsf{P}^{x_w}\mathsf{X}^{x_w}\mathsf{Z}^{z_w}\mathsf{T})\rho_w(\mathsf{P}^{x_w}\mathsf{X}^{x_w}\mathsf{Z}^{z_w}\mathsf{T})^\dagger.$$

  At this point, to correct the P^{x_w} error, Dulek et al. [DSS16] use the quantum gadget $\Gamma_{pk_{i+1}}(sk_i)$ that is given as input to the evaluator. In our scheme, the evaluator has to construct the quantum gadget itself from the classical evaluation key, using the evaluation key $gk_i, \tilde{t}_i^{[i+1]}, \tilde{sk}_i^{[i+1]}, pk_{i+1}, evk_{i+1}$,

  $$\Gamma_{pk_{i+1}}(sk_i) \leftarrow \mathsf{RSPGenGadget}\Big(gk_i, \tilde{t}_i^{[i+1]}, pk_{i+1}, evk_{i+1}\Big).$$

  The $\mathsf{RSPGenGadget}$ protocol is described later in this section. Now, as in DSS, the evaluator performs a sequence of measurements as specified by the function $\nu(\tilde{x}_w^{[i]}, \cdot)$. This has the effect of the qubit at wire w and

  $$(\mathsf{X}^{x'_w}\mathsf{Z}^{z'_w})\mathsf{T}\rho\mathsf{T}^\dagger(\mathsf{X}^{x'_w}\mathsf{Z}^{z'_w})^\dagger.$$

The evaluator also recrypts $\tilde{x}_w^{[i]}$ and $\tilde{y}_w^{[i]}$ under the $(i+1)$th key, to get $\tilde{x}_w^{[i+1]}$ and $\tilde{y}_w^{[i+1]}$. Using the measurement outcomes, the classical information part of the gadget $\Gamma_{pk_{i+1}}(sk_i)$ and the recryptions $\tilde{x}_w^{[i+1]}$, $\tilde{y}_w^{[i+1]}$ the evaluator homomorphically computes encryptions of the new keys $(\tilde{x}'_w)^{[i+1]}$, $(\tilde{y}'_w)^{[i+1]}$. After these computations, the evaluator also recrypts the keys of all the wires into the $(i+1)$th key set.

Decryption. As input, we are given the state $|\psi'\rangle$, which is hopefully close to the state $\mathsf{X}^x\mathsf{Z}^z C\,|\psi\rangle$, where C is the circuit and $C\,|\psi\rangle$ is the desired output state, except it is quantum one-time-padded with keys $x, z \in \{0,1\}^n$. We are also given encryptions $\tilde{x}, \tilde{z}$ of the one-time-pad keys x, z under pk_L.

$\mathsf{QFHE.Dec}_{sk}(|\psi'\rangle, \tilde{x}, \tilde{z})$

1. Decrypt encryptions of the one-time-pad keys using sk_L to get x, z.
2. Apply the Pauli correction $\mathsf{X}^x\mathsf{Z}^z$ to $|\psi'\rangle$ and output the resulting state.

5.1 Proof of Theorem 2

We first prove the quantum IND-CPA security of our scheme by constructing a sequence of hybrid schemes that transforms our scheme to the scheme by Broadbent and Jeffery [BJ15], which we will denote as BJ.

Lemma 8. *Suppose that* HE *is a IND-CPA-secure classical homomorphic encryption scheme. Then,* QFHE *is a quantum IND-CPA secure quantum homomorphic encryption scheme.*

Proof. Consider the following sequence of hybrid variations of our scheme. For all $\ell \in \{0,1,\ldots,L\}$, let S_ℓ be identical to our scheme except for the key generation procedure: for every $i \geq \ell$, $\mathsf{S}_\ell.\mathsf{KeyGen}$ replaces the output $gk_i(sk_i)$, $\mathsf{HE.Enc}_{pk_{i+1}}(t_{gk_i})$ of the subroutine call $\mathsf{GadgetKeyGen}(sk_i, pk_{i+1})$ as follows. The the gadget key $gk_i(sk_i)$ encoding the ith secret key is replaced with a gadget key $gk(0^{|sk_i|})$ encoding the all-zero string obtained by

$$gk(0^{|sk_i|}), \tilde{t} \leftarrow \mathsf{GadgetKeyGen}(0^{|\mathsf{sk_i}|}, \mathsf{pk_{i+1}})\ .$$

The encryption of t_{gk_i} under the public key pk_{i+1} is replaced with an encryption $\mathsf{HE.Enc}_{pk_{i+1}}(0^{|t_{gk_i}|})$ of the all-zero string $0^{|t_{gk_i}|}$ under pk_{i+1}.

Also, for every $\ell \in \{0, 1 \ldots L-1\}$, define scheme S'_ℓ be identical to $\mathsf{S}_{\ell+1}$, except that the encryption $\mathsf{HE.Enc}_{pk_{i+1}}(t_{gk_i})$ of the ith trapdoor under pk_{i+1} is replaced with an encryption of the all-zero string $0^{|t_{gk_i}|}$ under the same public key.

First, we claim that for any adversary $\mathcal{A}$, for all $\ell \in \{0,1,\ldots,L-1\}$, there exists a negligible function negl such that

$$\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_{\ell+1},\mathcal{A}\rangle}] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}'_\ell,\mathcal{A}\rangle}] \leq \mathsf{negl}(\lambda).$$

Otherwise, we can construct an adversary $\mathcal{A}'_{\mathsf{HE}}$ that breaks the IND-CPA security of the classical HE scheme by simulating the $\mathsf{G}^{\text{q-CPA}}_{\langle\rangle}$ game for $\mathcal{A}$.

Similarly, for any adversary $\mathcal{A}$, it must be the case that for all $\ell \in \{0,1,\ldots,L-1\}$, there exists a negligible function negl such that

$$\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}'_\ell,\mathcal{A}\rangle}] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_\ell,\mathcal{A}\rangle}] \leq \mathsf{negl}(\lambda).$$

Otherwise, we can construct an adversary $\mathcal{A}'_{\mathcal{F}}$ that breaks the mode indistinguishability property of the dual-mode trapdoor function family $\mathcal{F}$ by simulating $\mathsf{G}^{\text{q-CPA}}$ game for $\mathcal{A}$.

Note that S_L is exactly our scheme QFHE, and S_0 gives the adversary no extra information compared to the BJ scheme, so that for all adversaries $\mathcal{A}$,

$$\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_L, \mathcal{A}\rangle}(1^\lambda)] = \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{QFHE}, \mathcal{A}\rangle}(1^\lambda)], \text{ and}$$
$$\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_0, \mathcal{A}\rangle}(1^\lambda)] \leq \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{BJ}, \mathcal{A}\rangle}(1^\lambda)]$$

Therefore,

$$\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{QFHE}, \mathcal{A}\rangle}(1^\lambda)] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{BJ}, \mathcal{A}\rangle}(1^\lambda)] \leq \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_L, \mathcal{A}\rangle}(1^\lambda)] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_0, \mathcal{A}\rangle}(1^\lambda)] \tag{6}$$

We can write the right-hand side as a telescoping series

$$\begin{aligned}\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_L, \mathcal{A}\rangle}(1^\lambda)] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_0, \mathcal{A}\rangle}(1^\lambda)] &= \sum_{\ell=0}^{L-1} \Big(\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_{\ell+1}, \mathcal{A}\rangle}(1^\lambda)] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}'_\ell, \mathcal{A}\rangle}(1^\lambda)]\Big) \\ &\quad + \Big(\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}'_\ell, \mathcal{A}\rangle}(1^\lambda)] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{S}_\ell, \mathcal{A}\rangle}(1^\lambda)]\Big) \\ &\leq 2L \cdot \mathsf{negl}(\lambda) \leq \mathsf{negl}'(\lambda),\end{aligned}$$

for some other negligible function $\mathsf{negl}'(\cdot)$. By Equation (6), we know that there exists a neglible function negl' such that

$$\Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{QFHE}, \mathcal{A}\rangle}(1^\lambda)] - \Pr[\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{BJ}, \mathcal{A}\rangle}(1^\lambda)] \leq \mathsf{negl}'(\lambda).$$

By the quantum IND-CPA security of the BJ scheme, we get that no quantum polynomial-time adversaries $\mathcal{A}$ can win the quantum IND-CPA game $\mathsf{G}^{\text{q-CPA}}_{\langle \mathsf{QFHE}, \cdot\rangle}$ with respect to our scheme QFHE with more than negligible advantage. □

Proof of Theorem 2. Lemma 8 shows that the scheme QFHE is quantum IND-CPA secure. Lemma 7 shows the correctness of the protocol for preparing the DSS gadget on the evaluator side. Therefore, since the DSS scheme is correct and leveled fully homomorphic, so is ours. Finally, since our decryption scheme (which is identical to that of DSS) does not depend on the evaluated circuit, QFHE is compact. □

6 Instantiations

Our scheme in Section 5 allows us to build QFHE from generic classical FHE and dual-mode trapdoor functions. We now list the known instantiations of these primitives from various (concrete) cryptographic assumptions.

6.1 Instantiating post-quantum classical (leveled) FHE

Learning With Errors. The Learning With Errors assumption [Reg09] can be used to contruct both (leveled) fully homomorphic encryption as well as dual-mode trapdoor functions. Extensive research has been done on constructing FHE from LWE, culminating in the work of Brakerski and Vaikuntanathan [BV14] that builds FHE from the standard LWE assumption with polynomial modulus-to-noise ratio.

Ring Learning With Errors. Brakerski, Gentry and Vaikuntanathan also give an FHE scheme from Ring-LWE [BGV14], which can be directly imported into our QFHE scheme.

Post-Quantum FHE from Post-quantum Indistinguishability Obfuscation + perfectly re-randomizable encryption. The work of Canetti, Lin, Tessaro and Vaikuntanathan [CLTV15] obtains leveled classical FHE from subexponentially-secure IO and perfectly re-rerandomizable encryption. Their scheme is post-quantum as long as it is instantiated with subexponentially-secure post-quantum IO and post-quantum secure perfectly re-randomizable encryption. Although we have candidates for post-quantum IO [GGH15, BGMZ18, BDGM20, WW21], we do not have proofs of their post-quantum security under widely-believed assumptions. On the other hand, by an observation of Wichs [Wic24], there exists an El Gamal-style perfectly rerandomizable encryption scheme based on group actions.

6.2 Instantiating dTLFs

Post-Quantum dTF from Isogeny-based group action assumptions. Isogeny-based cryptography is a promising alternative to lattices for post-quantum cryptography. Initiated by Couveignes [Cou06] in 1997, it has led to new constructions of several cryptographic primitives such as collision-resistant hashing [CLG09], key exchange [RS06, Sto10], digital signatures [Sto09] and key escrow [Tes06]. More recent work has led to practical key exchange and public key encryption schemes, such as SIDH [JDF11, DFJP14] and CSIDH [CLM$^+$18]. Although the attack by Castryck and Decru [CD23] (classically) breaks SIDH, CSIDH is still plausibly (post-quantum) secure.

Alamati, De Feo, Montgomery and Patranabis [ADFMP20] present a framework based on group actions with natural cryptographic hardness properties to model isogeny-based assumptions and schemes.[9] This simplifying framework enables them to construct several new primitives, such as projective hashing, dual-mode PKE, two-message statistically sender-private OT, and a Naor-Reingold style PRF. They also introduce the Linear Hidden Shift assumption, based on which they construct symmetric KDM-secure encryption. Follow-up work of Alamati, Malavolta and Rahimi [AMR22] introduces the *Extended* Linear Hidden Shift assumption, and gives a construction of a (weak) trapdoor claw-free function from group actions under this assumption.

In Section 7, we extend their work to construct dual-mode trapdoor functions (with negligible correctness error) from the same assumption.

Learning With Errors (Superpolynomial modulus-to-noise ratio)

Mahadev [Mah18] constructs a dual-mode noisy trapdoor claw-free function family assuming the hardness of Learning With Errors (although with superpolynomial modulus). We slightly adapt this construction to match our definition of dTFs. Assuming the hardness of LWE for superpolynomial modulus-to-noise ratio, we obtain a dTF family.

Lemma 9. *Let λ be a security parameter. Let $q \geq 2$ be a prime. Let $n, m \geq 1$ be polynomially bounded functions of λ such that $m = \Omega(n \log q)$. Suppose that $LWE_{m,n,q,\rho_{\mathbb{Z}_q,\sigma}}$ is hard for superpolynomial modulus-to-noise ratio q/σ. Then, there exists an injective and invertible dual-mode trapdoor function family.*

[9] An important caveat is that the isogeny-based group actions have less-than-ideal algorithmic properties; for example, it is not always possible to efficiently compute the group action for any group element. One approach to fix this issue, taken by CSI-FiSh [BKV19], is to perform a preprocessing step and compute the group structure in the form of relation lattice of low-norm generators.

Proof. Let $B^* = q/(C\sqrt{mn\log q})$, where $C > 0$ is the universal constant as in Theorem 1. Let $B_1 = B^*/2$ and let $B_2 < B_1$ such that the ratio B_1/B_2 is superpolynomial in the security parameter λ.

Key Generation. The key $k = (\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ and trapdoor t_k is generated as follows. First $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ is generated along with a trapdoor $t_k = t_{\mathbf{A}}$ as in Theorem 1 [MP12]. In mode $\mu = 0$, $\mathbf{b} \leftarrow \mathbb{Z}_q^m$ is sampled uniformly at random. In mode mode $\mu = 1$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is sampled uniformly, and $\bar{\mathbf{e}} \leftarrow D_{\mathbb{Z}_q^m, B_2}$ and $\mathbf{b} = \mathbf{A}\mathbf{s} + \bar{\mathbf{e}}$.

Efficient Function Evaluation. The function $f_{k,b} : \mathbb{Z}_q^n \times \in \mathbb{Z}_q^m \to \mathbb{Z}_q^m$ takes as input $\mathbf{x} \in \mathbb{Z}_q^m$ and $\mathbf{e} \in \mathbb{Z}_q^m$ such that $\|\mathbf{e}\| \leq B^*\sqrt{m}$, and is defined as $f_{k,b}(\mathbf{x}, \mathbf{e}) := \mathbf{A}\mathbf{x} + b \cdot \mathbf{b} + \mathbf{e}$, which is efficiently computable given the key $k = (\mathbf{A}, \mathbf{b})$.

Efficient state preparation. Let $D = U(\mathbb{Z}_q^n) \otimes \rho_{\mathbb{Z}_q^m, B_1}$ be the product distribution of the uniform distribution on $\mathbb{Z}_q^n$ and the truncated discrete Gaussian distribution $\rho_{\mathbb{Z}_q^m, B_1}$. Let $D' = U(\mathbb{Z}_q^n) \otimes (\rho_{\mathbb{Z}_q^m, B_1} + \bar{\mathbf{e}})$, where $\rho_{\mathbb{Z}_q^m, B_1} + \bar{\mathbf{e}}$ is the *shifted* truncated discrete Gaussian distribution. By Lemma 3.12 of [Reg09], there is a quantum polynomial-time procedure that prepares the state

$$\sum_{\mathbf{x}\in\mathbb{Z}_q^n} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{D(\mathbf{x}, \mathbf{e})}\, |\mathbf{x}, \mathbf{e}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^n} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \frac{1}{q^{-n/2}} \cdot \sqrt{\rho_{\mathbb{Z}_q^m, B_1}(\mathbf{e})}\, |\mathbf{x}, \mathbf{e}\rangle \ ,$$

In mode $\mu = 0$, define $D_{k,0} = D_{k,1} = D$. In mode $\mu = 1$, for key $k = (\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \bar{\mathbf{e}})$, let $D_{k,0} = D$ and let $D_{k,1} = D'$. Although $\bar{\mathbf{e}}$ is not efficiently computable without the trapdoor, and therefore we cannot work with D' exactly, D is very close to D', and that will be good enough. Indeed since $\|\bar{\mathbf{e}}\| \leq B_2\sqrt{m}$, by Lemma 3 and the parameter setting, $H(D, D') \leq \mathsf{negl}(\lambda)$, and by Lemma 2, $\sum_{\mathbf{x}\in\mathbb{Z}_q^n} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{D(\mathbf{x}, \mathbf{e})}\, |\mathbf{x}, \mathbf{e}\rangle$ and $\sum_{\mathbf{x}\in\mathbb{Z}_q^n} \sum_{\mathbf{e}\in\mathbb{Z}_q^m} \sqrt{D'(\mathbf{x}, \mathbf{e})}\, |\mathbf{x}, \mathbf{e}\rangle$ are negligibly close in trace distance.

Injectivity and Efficient inversion with the trapdoor. Given the trapdoor and some $\mathbf{y} = \mathbf{A}\mathbf{x} + b\cdot\mathbf{b} + \mathbf{e}$ such that $\|\mathbf{e}\| \leq B_1\sqrt{m}$, we can use Theorem 1 for each $b \in \{0, 1\}$ on input $(\mathbf{A}, \mathbf{y} - b \cdot \mathbf{b})$ to recover $\mathbf{x}_b$ if it exists. This will give us all the possible preimages. This also immediately implies that $f_{k,b}$ is injective.

Dual-mode. In mode $\mu = 0$, with all but negligible probability there does not exist $\mathbf{s}, \bar{\mathbf{e}}$ such that $\mathbf{b} = \mathbf{A}\mathbf{s} + \bar{\mathbf{e}}$ such that $\|\bar{\mathbf{e}}\| \leq B^*\sqrt{m}$. Therefore there cannot exist $(\mathbf{x}_0, \mathbf{e}_0)$ and $(\mathbf{x}_1, \mathbf{e}_1)$ in the supports of $D_{k,0}$ and $D_{k,1}$ respectively such that $f_{k,0}(\mathbf{x}_0, \mathbf{e}_0) = f_{k,1}(\mathbf{x}_1, \mathbf{e}_1)$; otherwise, $\mathbf{A}\mathbf{x}_0 + \mathbf{e}_0 = \mathbf{A}\mathbf{x}_1 + \mathbf{b} + \mathbf{e}_1$. Rearranging, this means that $\mathbf{b} = \mathbf{A}(\mathbf{x}_0 - \mathbf{x}_1) + (\mathbf{e}_0 - \mathbf{e}_1)$, where $\|\mathbf{e}_0 - \mathbf{e}_1\| \leq \|\mathbf{e}_0\| + \|\mathbf{e}_1\| \leq 2B_1\sqrt{m} = B^*\sqrt{m}$, which is a contradiction.

In mode $\mu = 1$, $\mathbf{b} = \mathbf{A}\mathbf{s} + \bar{\mathbf{e}}$ for $\|\bar{\mathbf{e}}\| \leq B_2\sqrt{m}$, by the injectivity of $f_{k,b}$, there is a perfect matching $\mathcal{R}$ such that $((\mathbf{x}_0, \mathbf{e}_0), (\mathbf{x}_1, \mathbf{e}_1)) \in \mathcal{R}$ if and only if $f_{k,0}(\mathbf{x}_0, \mathbf{e}_0) = f_{k,1}(\mathbf{x}_1, \mathbf{e}_1)$. In particular, $f_{k,0}(\mathbf{x}_0, \mathbf{e}_0) = f_{k,1}(\mathbf{x}_1, \mathbf{e}_1)$ if and only if $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{s}$ and $\mathbf{e}_1 = \mathbf{e}_0 - \bar{\mathbf{e}}$. Therefore, $D_{k,0}(\mathbf{x}_0, \mathbf{e}_0) = D_{k,1}(\mathbf{x}_1, \mathbf{e}_1)$.

Mode indistinguishability. Assuming that the $\mathrm{LWE}_{n,q,\rho_{\mathbb{Z}_q,\sigma}}$ assumption is true for some superpolynomial modulus-to-noise ratio B_2/q, the keys in the two modes are computationally indistinguishable. □

Ring Learning With Errors

The dual-mode trapdoor function family from LWE described above can be directly translated to one from Ring-LWE. We now informally describe such a construction from ring-LWE. The [MP12] trapdoor sampling procedure readily generalizes to the Ring-LWE setting. Let $R_q := \mathbb{Z}_q[x]/(x^d+1)$ be a polynomial ring.

The key consists of two sequences of polynomials $k = (\mathbf{a}, \mathbf{u}) \in R_q^m \times R_q^m$. The sequence of polynomials $\mathbf{a} = (a_1, a_2, \ldots, a_m) \in R_q^m$ generated along with a trapdoor $t_{\mathbf{a}}$. In mode $\mu = 0$, $\mathbf{u} = (u_1, u_2, \ldots, u_m) \in R_q^m$ generated uniformly at random. In mode $\mu = 1$, $\mathbf{u} = \mathbf{a} \cdot s + \bar{\mathbf{e}}$, where $s \in R_q$ is any (worst-case) polynomial and $\bar{\mathbf{e}} \in R_q^m$ is a vector of polynomials whose each coefficient is sampled i.i.d. from a narrow-width discrete Gaussian. Then the function $f_{k,b}$ is defined as follows. For $x \in R_q$ and $\mathbf{e} \in R_q^m$,

$$f_{k,b}(x, \mathbf{e}) := \mathbf{a} \cdot x + b \cdot \mathbf{u} + \mathbf{e}.$$

Similar to the LWE case, in mode $\mu = 0$ the probability distributions $D_{k,0} = D_{k,1}$ are (wider) discrete Gaussians, and in mode $\mu = 1$, and the two distributions are discrete Gaussians of the same width, except that they are shifted by $\bar{\mathbf{e}}$, that is $D_{k,1} = D_{k,0} - \bar{\mathbf{e}}$.

Learning With Errors (Polynomial modulus-to-noise ratio)

We observe that we can also obtain (a q-ary generalization of) dual-mode trapdoor functions from the quantum hardness of LWE with a polynomial modulus-to-noise ratio from the work of Brakerski [Bra18], which builds a QFHE scheme from the same assumption. Unlike the dTF obtained from LWE with superpolynomial modulus-to-noise ratio, the functions we obtain here are not injective; however, there is still enough structure that allows us to correct the phase error introduced after the Hadamard measurement of the input register. We note that this q-ary variant of dTFs suffice for our purposes; instead of measuring the third register of $\left|\chi^{(4)}\right\rangle$ in the Hadamard basis in Lemma 6, we instead first take the q-ary QFT and then measure the third register in the standard basis.

The [Bra18] QFHE scheme uses the work of Bourse et al. [BDPMW16] that obtains circuit-private classical FHE from LWE with a polynomial modulus-to-noise ratio. In particular, we need a classical encryption scheme $\mathsf{HE}_{\mathsf{Bra18}}$ that supports a rerandomizable, circuit-private homomorphic scalar addition operation, with the following additional properties. The QFHE scheme of Brakerski [Bra18] implicitly relies on such an encryption, and instantiates it assuming the hardness of LWE with polynomial modulus-to-noise ratio. We informally describe the dTF construction and the properties we need from the encryption scheme.

Let P_0, P_1 be some distributions over the ciphertext space such that ciphertexts in the support of P_b decrypt to $b \in \{0, 1\}$, so $\mathrm{TV}(P_0, P_1) = 1$ by correctness of decryption. Further, let D be an efficiently sampleable distribution over finite domain $\mathcal{X}$.

1. There is a probabilistic polynomial-time key generation algorithm $pk, sk, evk \leftarrow \mathsf{HE}_{\mathsf{Bra18}}.\mathsf{KeyGen}(1^\lambda)$.

2. The encryption algorithm $c \leftarrow \mathsf{HE}_{\mathsf{Bra18}}.\mathsf{Enc}_{pk}(m; r)$ encrypts bits $m \in \{0, 1\}$ with the public key. The ciphertext space is contained in $\mathbb{Z}_q^n$ for some $q, n \in \mathbb{N}$.

3. There exists an algorithm that computes scalar addition as follows: $\mathsf{HE}_{\mathsf{Bra18}}.\mathsf{EvalAdd}_{evk}(c, b; x)$ takes as input an encryption c of $m \in \{0, 1\}$, a bit $b \in \{0, 1\}$ and randomness $x \leftarrow D$, and outputs ciphertext c' such that the distribution of c' (taken over the choice of $x \leftarrow D$) is negligibly close in Hellinger distance to $P_{m \oplus b}$.

4. There is an efficient quantum procedure that produces the state negligibly close to $\sum_{x\in\mathcal{X}} \sqrt{D(x)}\,|x\rangle$.
5. There is a decryption algorithm $\mathsf{HE}_{\text{Bra18}}.\mathsf{Dec}_{sk}$ that correctly decrypts the ciphertext outputs of $\mathsf{HE}_{\text{Bra18}}.\mathsf{Enc}$ and $\mathsf{HE}_{\text{Bra18}}.\mathsf{EvalAdd}$.

With such an encryption scheme $\mathsf{HE}_{\text{Bra18}}$, we can construct dTF as follows. The key generation procedure takes as input the security parameter 1^λ and the mode $\mu \in \{0,1\}$, and outputs the key-trapdoor pair $k = (k_0, k_1, evk)$ and $t_k = sk$, where $(pk, sk, evk) \leftarrow \mathsf{HE}_{\text{Bra18}}.\mathsf{KeyGen}(1^\lambda)$, $k_0 \leftarrow \mathsf{HE}_{\text{Bra18}}.\mathsf{Enc}_{pk}(0)$ and $k_1 \leftarrow \mathsf{HE}_{\text{Bra18}}.\mathsf{Enc}_{pk}(\mu)$. By the semantic security of $\mathsf{HE}_{\text{Bra18}}$, keys sampled from the two modes are computationally indistinguishable. The dTF is defined as

$$f_{k,b}(x) = \mathsf{HE}_{\text{Bra18}}.\mathsf{EvalAdd}_{evk}(k_b, b; x),$$

which is efficiently computable given the key $k = (k_0, k_1)$. In the disjoint mode ($\mu = 0$), the output of $f_{k,b}$ is always an encryption of the bit b, so $f_{k,0}$ and $f_{k,1}$ have disjoint images. In the lossy mode ($\mu = 1$), both $f_{k,0}$ and $f_{k,1}$ always output encryptions of 0. However these functions are no longer injective. To ensure the lossy mode property and efficient phase computation, we need the following additional property from the encryption scheme $\mathsf{HE}_{\text{Bra18}}$.

6. For all ciphertext outputs $y \in \mathbb{Z}_q^n$ of $\mathsf{HE}_{\text{Bra18}}.\mathsf{EvalAdd}$, and for all $d \in \mathbb{Z}_q^n$, it must be true that there is some $s \in \{0,1\}$ such that $\alpha_0 = (-1)^s \alpha_1$, where

$$\alpha_b = \sum_{\substack{x\in\mathcal{X} \\ f_{k,b}(x)=y}} e^{\frac{-2\pi i}{q}\langle d\cdot x\rangle} \cdot \sqrt{D(x)}.$$

   Further, given y, d as well as the secret key/trapdoor as input, there is an efficient algorithm that computes s for any y, d.

7 Dual-Mode Trapdoor Functions from Group Actions

Alamati, Malavolta and Rahimi give a candidate ε-weak trapdoor function from isogeny-based group actions [AMR22] for inverse-polynomial ε. For the security of their construction, they introduce and rely on the Extended Linear Hidden Shift (ELHS) assumption, a strengthening of the Linear Hidden Shift assumption of Alamati et al. [ADFMP20]. We modify their construction to give an *injective and invertible dual-mode trapdoor function* based on the ELHS assumption. Although this initially gives us a ε-weak dTF with the same inverse polynomial correctness error, Lemma 5 immediately implies a dTF with negligible correctness error. We start by setting up some notation and definitions.

Notation. For a regular and abelian group action $\star : \mathbb{G} \times \mathbb{X} \to \mathbb{X}$, we use additive notation to denote the group operation in $\mathbb{G}$. Let $\odot$ denote the component-wise product.

Definition 14 (Effective Group Action). *A regular and abelian 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 p.p.t. algorithms for:*
   (a) *membership testing (deciding whether a binary string represents a group element),*

   - (b) *equality testing and sampling uniformly in $\mathbb{G}$, and*
   - (c) *group operation and computing the inverse of any element.*

2. *The set $\mathbb{X}$ is finite and there exist efficient algorithms for*
   - (a) *membership testing (to check if a binary string represents a valid set element), and*
   - (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 $g \star x$.*

Isogeny-based group actions are believed to have cryptographic properties such as one-wayness and pseudorandomness, suitably defined. One such conjectured hardness property is captured by the Extended Linear Hidden Shift assumption, introduced by Alamati, Malavolta and Rahimi [AMR22]. We note that a single-copy version of the assumption as stated in [AMR22] suffices for us. Alamati et al. [AMR22] need (something stronger than) the multiple-copy version, to construct a trapdoor claw-free function with the adaptive hardcore bit property, which we do not require. We now define the single-copy version of their assumption.

Definition 15 (Single-copy Extended Linear Hidden Shift Assumption [AMR22])**.** *Let λ be the security parameter. Let $(\mathbb{G}, \mathbb{X}, \star)$ be an effective group action, and let $n \geq \log(|\mathbb{G}|) + \Omega(\operatorname{poly}(\lambda))$ be an integer. We say that the extended linear hidden shift (LHS) assumption holds over $(\mathbb{G}, \mathbb{X}, \star)$ if no quantum polynomial-time adversary can distinguish between the following two distributions with advantage greater than negligible in λ:*

$$D_0 := \big(\mathbf{M}, \mathbf{m}, (\mathbf{a}_b, \mathbf{b}_b)_{b \in \{0,1\}}\big) \quad \textit{and} \quad D_1 := \big(\mathbf{M}, \mathbf{m}, (\mathbf{x}_b, \mathbf{y}_b)_{b \in \{0,1\}}\big) \ , \tag{7}$$

where $\mathbf{M} \leftarrow \mathbb{G}^{n \times n}$, $\mathbf{m} \leftarrow \mathbb{G}^n$, $\mathbf{s} \leftarrow \{0,1\}^n$, $\mathbf{t} \leftarrow \mathbb{G}^n$, $\mathbf{a}_b \leftarrow \mathbb{X}^n$, and $\mathbf{b}_b \leftarrow \mathbb{X}^n$ for $b \in \{0,1\}$, and

$$\begin{aligned}
&\mathbf{x}_0 \leftarrow \mathbb{X}^n, \\
&\mathbf{y}_0 := \mathbf{t} \star \mathbf{x}_0, \\
&\mathbf{x}_1 := [\mathbf{M}\mathbf{s}] \star \mathbf{x}_0, \\
&\mathbf{y}_1 := [\mathbf{M}\mathbf{s} + \mathbf{m} \odot \mathbf{s}] \star \mathbf{y}_0 \ .
\end{aligned}$$

We make the following assumption regarding the structure of the group $\mathbb{G}$. This allows us to reason about the injectivity of the dTF family we construct. Informally, we require that the group order $\operatorname{ord}(\mathbb{G})$ has large prime factors. This assumption seems to be heuristically reasonable if we look at parameters for CSIDH-512, for which the group $\mathbb{G}$ is isomorphic to $\mathbb{Z}/N\mathbb{Z}$ where N is

$$\begin{aligned}
N = 3 \cdot 37 \cdot 1407181 \cdot 51593604295295867744293584889 \\
\cdot 31599414504681995853008278745587832204909 \approx 2^{257.1},
\end{aligned}$$

and its largest prime factor is roughly $2^{134.5}$ [BKV19].

Assumption 1. *Let $\{\mathbb{G}_\lambda\}_{\lambda \in \mathbb{N}}$ be a collection of finite abelian group with λ being the security parameter such that the following is true. Suppose the prime factorization of the group order is $\operatorname{ord}(\mathbb{G}_\lambda) = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ where $p_1 \leq \ldots \leq p_k$. Then, it must hold that $p_k = \lambda^{\omega(1)}$.*

Lemma 10. *Suppose the single-copy Extended Linear Hidden Shift Assumption holds over $(\mathbb{G}, \mathbb{X}, \star)$ with parameters λ and $n \leq \operatorname{poly}(\lambda)$. Further suppose that Assumption 1 holds for $\mathbb{G}$. Then, there exists an injective and invertible ε-weak dual-mode trapdoor function family, where $\varepsilon \leq 1/\operatorname{poly}(\lambda)$.*

By Lemma 5, we immediately get the following corollary.

Corollary 1. *Suppose the single-copy Extended Linear Hidden Shift Assumption holds over $(\mathbb{G}, \mathbb{X}, \star)$ with parameters λ and $n \leq \operatorname{poly}(\lambda)$, and that Assumption 1 holds for $\mathbb{G}$. Then, there exists an ε-weak dual-mode trapdoor function family, where $\varepsilon \leq 1/\mathsf{negl}(\lambda)$.*

Proof of Lemma 10. Let B be an integer such that $B > 2n^2$ and $B = \operatorname{poly}(\lambda)$.[10]

Efficient key generation. The key-generation algorithm $\mathsf{Gen}(1^\lambda, \mu)$ works as follows.

$\underline{\mathsf{Gen}(1^\lambda, \mu)}$:

1. Sample $\mathbf{M} \leftarrow \mathbb{G}^{n \times n}$, $\mathbf{m} \leftarrow \mathbb{G}^n$, $\mathbf{s} \leftarrow \{0,1\}^n$, and $\mathbf{t} \leftarrow \mathbb{G}^n$.
2. Sample $\mathbf{u}, \mathbf{v} \leftarrow \mathbb{G}^n$ and $\mathbf{x}_0 \leftarrow \mathbb{X}^n$ and set $\mathbf{x}_0, \mathbf{x}_1, \mathbf{y}_0, \mathbf{y}_1$ as follows:
$$\begin{aligned} \mathbf{y}_0 &:= \mathbf{t} \star \mathbf{x}_0, \\ \mathbf{x}_1 &:= [\mathbf{M}\mathbf{s} + (1-\mu) \cdot \mathbf{u}] \star \mathbf{x}_0, \\ \mathbf{y}_1 &:= [\mathbf{M}\mathbf{s} + \mathbf{m} \odot \mathbf{s} + (1-\mu) \cdot \mathbf{v}] \star \mathbf{y}_0. \end{aligned}$$
3. Set the key-trapdoor pair as
$$\begin{aligned} k &= (\mathbf{M}, \mathbf{m}, \mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_1, \mathbf{y}_1), \quad \text{and} \\ t_k &= \Big(\mathbf{s}, \mathbf{t}, (1-\mu) \cdot (\mathbf{u} - \mathbf{v})\Big). \end{aligned}$$

Efficient function evaluation. Define a family of functions where for each k, b, we have that $f_{k,b} : [B]^n \to \mathbb{X}^{2n}$ is defined as
$$f_{k,b}(\mathbf{r}) = \left(\mathbf{z}_{k,b}(\mathbf{r}), \mathbf{z}'_{k,b}(\mathbf{r})\right),$$
where
$$\begin{aligned} \mathbf{z}_{k,b}(\mathbf{r}) &:= [\mathbf{M}\mathbf{r}] \star \mathbf{x}_b, \quad \text{and} \\ \mathbf{z}'_{k,b}(\mathbf{r}) &:= [\mathbf{M}\mathbf{r} + \mathbf{m} \odot \mathbf{r}] \star \mathbf{y}_b\ . \end{aligned}$$

Assuming the group operation as well as the group action is efficiently computable, it is clear from the definition that the function can be efficiently evaluated give the key.

Efficient state preparation. Let the distribution $D_{k,b}$ be the uniform distribution over $[B]^n$ for all k, b, so there is an efficient quantum procedure to prepare the state $\sum_{\mathbf{r} \in [B]^n} D_{k,b}(\mathbf{r}) \left|\mathbf{r}\right\rangle$.

[10] We have some flexibility in terms of setting the parameters: even a lower bound of just $B \geq 4n$ gives us a $1/2$-weak dTF, which we can amplify to get negligible correctness error using Lemma 5.

Efficient inversion with trapdoor, and injectivity. To invert the function $f_{k,b}$ on some output $(\mathbf{z}, \mathbf{z}') = f_{k,b}(\mathbf{r})$, we can recover input $\mathbf{r}$ as follows. Observe that the following relation holds:

$$[-\mathbf{t} - b \cdot (1-\mu) \cdot (\mathbf{u} - \mathbf{v})] \star \mathbf{z}' = [\mathbf{m} \odot (\mathbf{r} + b \cdot \mathbf{s})] \star \mathbf{z} \,.$$

Because the action on the right-hand side is applied component-wise and each entry of $\mathbf{r}$ can take on at most polynomially many values in $[B]$, we can recover each entry of $\mathbf{r}$ efficiently by brute force, since $\mathbf{s}$, $\mathbf{t}$ and $(1-\mu) \cdot (\mathbf{u} - \mathbf{v})$ are included in the trapdoor. By Lemma 11, we know that with all but negligible probability, every element of $\mathbf{m}$ has order larger than B, therefore, $\mathbf{r}$ is uniquely determined.

Dual-mode. In the disjoint mode ($\mu = 0$) we claim that with all but negligible probability, the images of $f_{k,0}$ and $f_{k,1}$ are disjoint. Suppose for contradiction that there exist $\mathbf{r}, \mathbf{r}' \in [B]^n$ such that $f_{k,0}(\mathbf{r}) = f_{k,1}(\mathbf{r}')$. In particular, $z_{k,0}(\mathbf{r}) = z_{k,1}(\mathbf{r}')$. That is,

$$[\mathbf{M}\mathbf{r}] \star \mathbf{x}_0 = [\mathbf{M}\mathbf{r}'] \star \mathbf{x}_1 = [\mathbf{M}(\mathbf{r}' + \mathbf{s}) + \mathbf{u}] \star \mathbf{x}_0.$$

Since the group action is regular, we get that $\mathbf{M}\mathbf{r} = \mathbf{M}(\mathbf{r}' + \mathbf{s}) + \mathbf{u}$, or

$$\mathbf{M}(\mathbf{r} - \mathbf{r}' - \mathbf{s}) = \mathbf{u}.$$

Varying $\mathbf{v}, \mathbf{v}', \mathbf{s}$, the left hand side can take at most $(2B+1)^n$ values. On the other hand, $\mathbf{u}$ is sampled uniformly at random from a set of size $|\mathbb{X}|^n$. Therefore, the probability over the choice of the key k that there exist such values of $\mathbf{r}, \mathbf{r}'$ is at most $((2B+1)/|\mathbb{X}|)^n$, which is negligible in λ.

In the lossy mode ($\mu = 1$), we recover (a simpler version of) the weak TCF of [AMR22]. Observe that for any secret vector $\mathbf{s} \in \{0, 1\}$, with probability at least $1 - n/(B-1)$ over the choice of $\mathbf{r} \leftarrow [B^n]$, $\mathbf{r}' := \mathbf{r} - \mathbf{s}$ is also in the domain $[B]^n$. By injectivity, this means that $|\alpha_{k,y,d}(0)| = |\alpha_{k,y,d}(1)|$ with the same probability. By the choice of parameters, $\varepsilon = n/(B-1) \leq 1/\operatorname{poly}(\lambda)$.

Mode indistinguishability. In mode $\mu = 0$, since $\mathbf{u}, \mathbf{v}$ are sampled independently and uniformly, $\mathbf{x}_0, \mathbf{x}_1, \mathbf{y}_0, \mathbf{y}_1$ are i.i.d. uniform random vectors, and therefore $k(0)$ follows the distribution D_0 in Equation (7). On the other hand, in mode $\mu = 1$ the distribution of the key $k(1)$ is exactly D_1. Therefore the Extended Linear Hidden Shift assumption directly implies mode indistinguishability. □

Lemma 11. *Suppose Assumption 1 holds for the collection of groups $\{\mathbb{G}_\lambda\}_{\lambda \in \mathbb{N}}$, and let $B = \operatorname{poly}(\lambda)$ be an integer. Then, for a random element $g \leftarrow \mathbb{G}$, we have that $\operatorname{ord}(g) \geq B$ with probability at least $1 - \mathsf{negl}(\lambda)$.*

Proof. Since $\mathbb{G}$ is a finite abelian group, we can decompose $\mathbb{G}$ as a direct sum of cyclic groups $A_{i,j} := \mathbb{Z}_{p_i^{\beta_{i,j}}}$ of prime-power order, where $j \in [k_i]$, $\beta_{i,j} \geq 1$ and $\sum_j \beta_{i,j} = \alpha_i$. Sampling a uniformly random group element $g \leftarrow \mathbb{G}$ is equivalent to sampling its components $g_{i,j} \leftarrow A_{i,j}$ for all $i \in [k]$ and $j \in [k_i]$.

Observe that $\operatorname{ord}(g) \geq \max_{i,j} \operatorname{ord}(g_{i,j})$. In particular, consider a component cyclic group, say $A_{k,1}$, whose order is a power of the largest prime p_k. Since every non-identity element of this component cyclic group has order at least p_k, with probability at least $1 - p_k^{-\beta_{k,1}} = 1 - \lambda^{-\omega(1)}$, a random element $g_{k,1} \leftarrow A_{k,1}$ has order $\operatorname{ord}(g_{k,1}) \geq p_k$. Since $B = \operatorname{poly}(\lambda) \leq p_k$, this completes the proof. □

Acknowledgements. This research was supported in part by DARPA under Agreement Number HR00112020023, NSF CNS-2154149, a Simons Investigator award and a Thornton Family Faculty Research Innovation Fellowship. AG was supported in addition by the Ida M. Green Fellowship from MIT.

References

[ADFMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26*, pages 411–439. Springer, 2020. 29, 32

[AMR22] Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In *Theory of Cryptography Conference*, pages 266–293. Springer, 2022. 4, 29, 32, 33, 35

[Bar86] David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 1–5, 1986. 39

[BCM$^+$21] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. *Journal of the ACM (JACM)*, 68(5):1–47, 2021. 3

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. *Cryptology ePrint Archive*, 2020. 29

[BDPMW16] Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee. Fhe circuit privacy almost for free. In *Annual International Cryptology Conference*, pages 62–89. Springer, 2016. 31

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of ggh15: provable security against zeroizing attacks. In *Theory of Cryptography Conference*, pages 544–574. Springer, 2018. 29

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory*, 6(3):13:1–13:36, 2014. 29

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In *Annual Cryptology Conference*, pages 609–629. Springer, 2015. 4, 7, 9, 10, 19, 27

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. Csi-fish: efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 227–247. Springer, 2019. 29, 33

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 67–95. Springer, 2018. 3, 4, 7, 11, 31

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014. 28

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 423–447. Springer, 2023. 29

[CLG09] Denis X Charles, Kristin E Lauter, and Eyal Z Goren. Cryptographic hash functions from expander graphs. *Journal of CRYPTOLOGY*, 22(1):93–113, 2009. 29

[CLM^{+}18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. Csidh: an efficient post-quantum commutative group action. In *Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24*, pages 395–427. Springer, 2018. 29

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12*, pages 468–497. Springer, 2015. 3, 4, 29

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. *Cryptology ePrint Archive*, 2006. 29

[DFJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014. 29

[DK16] Vedran Dunjko and Elham Kashefi. Blind quantum computing with two almost identical states. *arXiv preprint arXiv:1604.01586*, 2016. 7

[DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III 36*, pages 3–32. Springer, 2016. 4, 5, 7, 19, 23, 25, 26, 39, 40, 41

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II 12*, pages 498–527. Springer, 2015. 29

[GMP22] Alexandru Gheorghiu, Tony Metger, and Alexander Poremba. Quantum cryptography with classical communication: parallel remote state preparation for copy-protection, verification, and more. *arXiv preprint arXiv:2201.13445*, 2022. 7

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013. 3

[GV19] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1024–1033. IEEE, 2019. 7

[JDF11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4*, pages 19–34. Springer, 2011. 29

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018. 3, 11, 29

[Mah20] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. *SIAM Journal on Computing*, (0):FOCS18–189, 2020. 3, 4, 7, 11

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012. 11, 30, 31

[PRSD17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 461–473, 2017. 10

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009. 10, 28, 30

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *Cryptology ePrint Archive*, 2006. 29

[Sto09] Anton Stolbunov. Reductionist security arguments for public-key cryptographic schemes based on group action. *Norsk informasjonssikkerhetskonferanse (NISK)*, pages 97–109, 2009. 29

[Sto10] Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. Math. Commun.*, 4(2):215–235, 2010. 29

[SV03] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM (JACM)*, 50(2):196–249, 2003. 17

[Tes06] Edlyn Teske. An elliptic curve trapdoor system. *Journal of cryptology*, 19:115–133, 2006. 29

[Wic24] Daniel Wichs. Rerandomizable encryption from group actions. Personal Communication, 2024. 4, 29

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious lwe sampling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 127–156. Springer, 2021. 29

A The DSS Teleportation Gadget

In this section, we describe the structure of the gadget and how the evaluator uses the gadget to correct the P gate errors introduced when performing a T gate. As depicted in Figure 1, the procedure of running the DSS gadget takes as input a state ρ and an encryption $\tilde{x} = \mathsf{HE.Enc}_{pk}(x)$ for some $x \in \{0,1\}$ and outputs the state

$$\mathsf{X}^{x'}\mathsf{Z}^{z'}(\mathsf{P}^\dagger)^x \rho (\mathsf{P})^x \mathsf{Z}^{z'}\mathsf{X}^{x'},$$

for some one-time pad keys $x', z' \in \{0,1\}$, along with their encryptions $\mathsf{HE.Enc}_{pk'}(x')$ and $\mathsf{HE.Enc}_{pk'}(z')$ under an independent public key pk'. At a high level, what the gadget decrypts $\tilde{x}$ and then applies $(\mathsf{P}^\dagger)^x$ to the input qubit in a way that does not reveal the secret key sk. Dulek et al. [DSS16] use the fact that the decryption circuit is in NC^1, that is, its depth grows logarithmically with the security parameter, to write a polynomial-size width-5 permutation branching program $\mathcal{P}$ that computes $\mathsf{HE.Dec}$ (via Barrington's theorem [Bar86]). The permutation branching program can then be encoded as Bell pairs that wired together in a particular manner. Teleporting an input qubit through these Bell pairs has the effect of performing the decryption computation.

The construction. Now we present the construction of the DSS gadget explicitly, starting with the permutation branching program we want to encode. For intuition, see Example 1 and Figure 4 taken from [DSS16] for a picture of what the DSS gadget would look like if the decryption circuit was simply an OR of the first bits of the ciphertext and the secret key. Suppose that the decryption circuit $\mathsf{HE.Dec}$ takes as input the concatenation of the secret key sk and the ciphertext $\tilde{x}$, and has depth $O(\log(\lambda))$, where λ is the security parameter. Then, by Barrington's theorem, we can write $\mathsf{HE.Dec}$ as a width-5 permutation branching program $\mathcal{P}$ of length $L = \mathrm{poly}(\lambda)$,

$$\mathcal{P} = (\langle i_1, \sigma_1^0, \sigma_1^1\rangle, \langle i_2, \sigma_2^0, \sigma_2^1\rangle, \ldots, \langle i_L, \sigma_L^0, \sigma_L^1\rangle).$$

Without loss of generality, we can assume that L is even, and that the instructions alternately depend on bits from the ciphertext $\tilde{x}$ and the secret sk. This can be ensured by inserting dummy instructions always performing the identity permutation between any two consecutive instructions that depend on the same variable (either sk or $\tilde{x}$). Suppose that the ℓth instruction $\langle i_\ell, \sigma_\ell^1, \sigma_\ell^0\rangle$ depends on $\tilde{x}$ if ℓ is odd and sk if ℓ is even.

The DSS gadget encodes the even-numbered instructions using entanglement between Bell pairs and Bell basis measurements to encode the odd-numbered instructions. The gadget is arranged in layers, and teleporting the input qubit through the DSS gadget implicitly runs $\mathsf{HE.Dec}_{sk}$ on $\tilde{x}$, where each layer effectively applies a permutation in S_5 that teleports it to one of the five registers in the next layer.

1. Let Q be a set of $10L$ qubit registers. Label the first $5L$ of them in layers of ten as $1_{\ell,\text{in}}, 2_{\ell,\text{in}}, \ldots, 5_{\ell,\text{in}}$ and $1_{\ell,\text{out}}, 2_{\ell,\text{out}}, \ldots, 5_{\ell,\text{out}}$ for *even* $\ell \in [2L]$.
2. Let Q_1 be the set of registers labeled as $1_{\ell,\text{in}}, 2_{\ell,\text{in}}, \ldots, 5_{\ell,\text{in}}$ for even $\ell \le 2L$, and Q_2 be the set of registers labeled as $1_{\ell,\text{out}}, 2_{\ell,\text{out}}, \ldots, 5_{\ell,\text{out}}$ for *even* $\ell \in [2L]$.
3. Let $P = \{2_{L,\text{out}}, 3_{L,\text{out}}, \ldots, 5_{L,\text{out}}\}$ be the set of registers with the inverse phase gate applied to them.
4. The Bell pairs between Q_1 and Q_2 are determined the even-numbered instructions $\langle i_\ell, \sigma_\ell^1, \sigma_\ell^0\rangle$, for even $\ell \in [L]$.[11] In particular, if $a_{\ell,\text{in}} \in Q_1$ for some $a \in [5]$, then $\pi^{sk}(a_{\ell,\text{in}}) = \sigma_\ell(a)_{\ell,\text{out}}$, where

[11] For now we only describe the behavior of π (and ν) on the first $5L/2$ qubits of Q_1 (and Q_2).

$\sigma_\ell := \sigma_\ell^{sk_{i_\ell}}$ is determined by i_ℓth bit of sk.

5. The Bell basis measurements to be performed by the evaluator depend on the ciphertext $\tilde{x}$ and are determined by the odd-numbered instructions $\langle i_\ell, \sigma_\ell^1, \sigma_\ell^0 \rangle$, for odd $\ell \in [L]$. If $a_{\ell-1,\text{out}} \in Q_2$ for some $a \in [5]$, then $\nu(a_{\ell,\text{out}}) = \sigma_\ell(a)_{\ell,\text{in}}$, where $\sigma_\ell := \sigma_\ell^{sk_{i_\ell}}$. Labeling the input qubit as $1_{0,\text{out}}$, we also have that $\nu(1_{0,\text{out}}, \tilde{x}) = \sigma_1(1)_{1,\text{in}}$.

By Barrington's theorem, if $\mathsf{HE.Dec}_{sk}(\tilde{x}) = 0$ then the product, say τ, of the permutations coming from the evaluated instructions equals the identity permutation. In that case, consecutively applying these permutations on '1' results in $\tau(1) = e(1) = 1$. The teleported qubit ends up in the register $1_{L,\text{out}}$ (with no inverse phase gate). On the other hand, if $\mathsf{HE.Dec}_{sk}(\tilde{x}) = 1$ then τ would evaluate to a 5-cycle, and consecutively applying these permutations on '1' results in $\tau(1) \in \{2, 3, 4, 5\}$. In this case, the teleported qubit ends up in one of registers $2_{L,\text{out}}, 3_{L,\text{out}}, \ldots, 5_{L,\text{out}}$, with an inverse phase gate applied to it.

The gadget so far correctly applies the inverse phase gate $\mathsf{P}^\dagger$ if and only if $\mathsf{HE.Dec}_{sk}(\tilde{x}) = x = 1$. However, the qubit is at a location that is unknown to the evaluator, because τ is hidden from them! To fix this problem, [DSS16] use the remaining $5L$ qubits to execute the inverse branching program afterwards, so that the desired qubit is always teleported to the same register. The inverse branching program, which we denote as $\mathcal{P}^{-1}$ is given by the instructions of the original program $\mathcal{P}$ in reverse order, using the inverse permutations for each instruction instead,

$$\mathcal{P}^{-1} = \Big(\langle i_L, (\sigma_L^1)^{-1}, (\sigma_L^0)^{-1} \rangle, \langle i_{L-1}, (\sigma_{L-1}^1)^{-1}, (\sigma_{L-1}^0)^{-1} \rangle, \ldots, \langle i_1, (\sigma_1^0)^{-1}, (\sigma_1^0)^{-1} \rangle \Big).$$

After executing the entire gadget with $\mathcal{P}^{-1}$, the qubit is guaranteed to be in the location it started in: register '1' of the final layer of five qubits[12]. This completes the description of the DSS gadget.

Example 1. *As a simplified example, suppose that $sk, \tilde{x} \in \{0, 1\}$ the decryption function $\mathsf{HE.Dec}_{sk}(\tilde{a})$ is sk OR $\tilde{x}$. Then, for one possible example set of values of $\tilde{a}$ and sk, half of the gadget and measurements will be as given in Figure 4. To complete this gadget, the same construction is appended, reflected horizontally. The OR function on two bits can be computed using a width-5 permutation branching program of length 4, consisting of the following list of instructions:*

1. $\langle 1, e, (12345) \rangle$
2. $\langle 2, e, (12453) \rangle$
3. $\langle 1, e, (54321) \rangle$
4. $\langle 2, (14235), (15243) \rangle$

[12] Note that instruction L is used twice in a row, breaking the alternation between 'in' and 'out' registers. This can be fixed by performing measurements that correspond to the identity permutation in between.

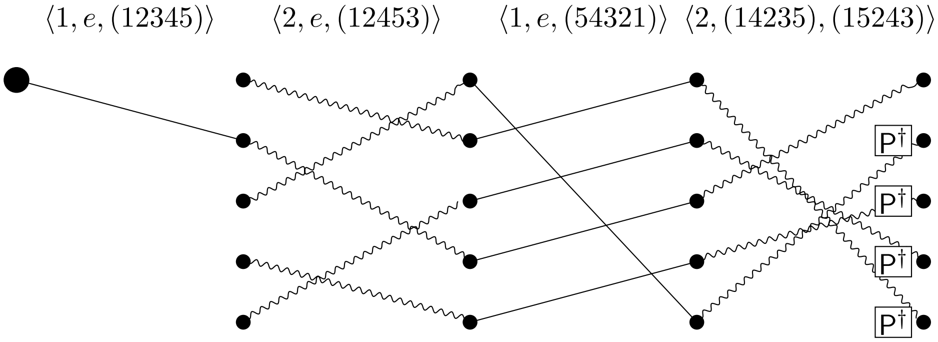


Figure 4: (Taken from [DSS16], Figure 4): Structure of the (first half of the) gadget, with measurements, coming from the 5-permutation branching program for the OR function on the input $(sk, \tilde{x}) = (0, 0)$. The example program's instructions are displayed above the permutations. The solid lines correspond to Bell measurements, while the wavy lines represent EPR pairs.