Secure Software Leasing Without Assumptions

Anne Broadbent[1], Stacey Jeffery[2], Sébastien Lord[1], Supartha Podder[1], and Aarthi Sundaram[3]

[1] University of Ottawa, Ottawa, Canada.
{abroadbe,slord050,spodder}@uottawa.ca
[2] QuSoft and CWI, Amsterdam, Netherlands. jeffery@cwi.nl
[3] Microsoft Quantum, Redmond, USA. aarthi.sundaram@microsoft.com

Abstract. Quantum cryptography is known for enabling functionalities that are unattainable using classical information alone. Recently, *Secure Software Leasing (SSL)* has emerged as one of these areas of interest. Given a target circuit C from a circuit class, SSL produces an encoding of C that enables a recipient to evaluate C, and also enables the originator of the software to *verify* that the software has been *returned* — meaning that the recipient has relinquished the possibility of any further use of the software. Clearly, such a functionality is unachievable using classical information alone, since it is impossible to prevent a user from keeping a copy of the software. Recent results have shown the achievability of SSL using quantum information for a class of functions called *compute-and-compare* (these are a generalization of the well-known *point functions*). These prior works, however all make use of setup or computational assumptions. Here, we show that SSL is achievable for compute-and-compare circuits *without any assumptions*.

Our technique involves the study of *quantum copy protection*, which is a notion related to SSL, but where the encoding procedure inherently *prevents* a would-be quantum software pirate from *splitting* a single copy of an encoding for C into two parts, each of which enables a user to evaluate C. We show that point functions can be copy-protected *without any assumptions*, for a novel security definition involving one honest and one malicious evaluator; this is achieved by showing that from any quantum message authentication code, we can derive such an *honest-malicious* copy protection scheme. We then show that a generic honest-malicious copy protection scheme implies SSL; by prior work, this yields SSL for compute-and-compare functions.

1 Introduction

One of the defining features of quantum information is the *no-cloning* principle, according to which it is not possible, in general, to take an arbitrary quantum state and produce two copies of it [13, 22, 25]. This principle is credited for many of the feats of quantum information in cryptography, including quantum key distribution (QKD) [7] and quantum money [24]. (For a survey on quantum cryptography, see [9]). The quantum no-cloning principle tells us that, in a

certain sense, quantum information behaves more like a *physical* object than a digital one: there are situations where quantum information can be distributed and used, but it cannot be duplicated. One such example is quantum money [24], in which a quantum system is used to encode a very basic type of information — the ability to verify authenticity. However, we can envisage quantum encodings that achieve richer levels of applicability. We thus define a hierarchy of "uncloneable" objects, where the basic notion provides only authenticity, and the topmost notion provides *functionality*. The uncloneability hierarchy includes:

- **Authenticity.** In the first (most basic) level, the uncloneability property can be used to *verify* authenticity.
- **Information.** Next, *information* is made uncloneable, meaning that there is some underlying data that can be decoded, but there are limitations on the possibility of copying this data while it is encoded.
- **Functionality.** At the top level of the hierarchy, a *functionality* is made uncloneable, meaning that there are limitations on how many users can simultaneously evaluate the functionality.

For both the case of *information* and *functionality*, a type of *verification* is possible (but optional): this verification is a way to confirm that a message or functionality is returned; after such verification is confirmed, further reading/use of the encoded information is impossible.

We emphasize that none of the concepts in the hierarchy are possible in a conventional digital world, since classical information can be copied. Thus the hierarchy is best understood intuitively at the level of a physical analogy where, for example, authenticity is verified by physical objects and functionalities are distributed in *hardware* devices.

Achieving the hierarchy. We summarize below the known results on achievability of the hierarchy.

1. The *authenticity* level of the hierarchy is the most well-understood, and it includes quantum money [24], quantum coins [20], and publicly-verifiable quantum money [2].
2. Next, the *information* level includes *tamper-evident* encryption [18] and *uncloneable encryption* [8]. We comment here on a technique of Gottesman [18] that is relevant to our work. In [18], it is shown that tamper-evident encryption can be achieved using the primitive of *Quantum Message Authentication (QMA)* [6] — in other words, the *verification* of quantum authentication not only gives a guarantee that the underlying plaintext is intact, but *also* that no adversary can gain information on the plaintext, *even if the key is revealed*. Uncloneable encryption is a notion that is complementary to tamper-evident encryption, and it focuses on *preventing* duplication of an underlying plaintext. In [8], it is shown to be achievable in the Quantum Random Oracle Model (QROM).
3. Finally, the *functionality* level of the hierarchy was first discussed in terms of *quantum copy protection* by Aaronson [1]: here, a quantum encoding allows

the evaluation of a function on a chosen input, but in a way that the number of *simultaneous* evaluations is limited. In [1], copy protection for a class of functions is shown to exist assuming a quantum oracle; this was improved (for a more restricted family of circuits) to a *classical* oracle in [3]. Further work in [11] improved the assumption to the QROM.[4]
A related concept, also at the functionality level of the hierarchy, was recently put forward: *Secure Software Leasing (SSL)*, where a quantum encoding allows evaluation of a circuit, while also enabling the originator to verify that the software is *returned* (meaning that it can no longer be used to compute the function). SSL was first studied by Ananth and La Placa [5][5] where it was shown that SSL could be achieved for *searchable compute-and-compare circuits*[6]; in order to achieve their result (which is with respect to an *honest* evaluation), they make use of strong cryptographic assumptions: quantum-secure subspace obfuscators, a common reference string, and the difficulty of the Learning With Errors (LWE) problem. Further work [11] improved the result on achievability for the same class of circuits, this time against *malicious evaluations*, and in the QROM. Very recently, [19] showed the achievability of SSL, based on LWE, against honest evaluators, and for classes of functions beyond *evasive* functions.[7]

1.1 Summary of Contributions

Due to their foundational role in the study of uncloneability as well as for potential applications, SSL and copy protection are emerging as important elements of quantum cryptography. In this work, we solve two important open problems related to SSL and quantum copy protection.

Secure Software Leasing. We show how to construct an SSL scheme for compute-and-compare circuits, against a malicious evaluator. Ours is the first scheme that makes no assumptions — there are no setup assumptions, such as the QROM or a common reference string and no computational assumptions, such as one-way functions or the LWE assumption. We thus show for the first time that SSL is achievable, unconditionally. A compromise we make in order to achieve this is the

[4] This is an improvement, as a QROM does not depend on the circuit to be computed.
[5] Two notions are actually introduced in [5]: *finite-term* and *infinite-term* SSL. In this work, SSL refers to finite-term SSL. Furthermore, in [5] all the evaluators in the security game are assumed to behave honestly. In this work, we do not make this assumption and our SSL evaluators can behave maliciously.
[6] A circuit class $\mathcal{C}$ is a *compute-and-compare* circuit class if for every circuit in $\mathcal{C}$, there is an associated circuit C and string α such that on input x, the circuit outputs 1 if and only if $C(x) = \alpha$. *Searchability* refers to the fact that there is an efficient algorithm that, on input $C \in \mathcal{C}$, outputs an x such that $C(x) = \alpha$. From this point on, *searchability* is an implicit assumption throughout this work.
[7] Informally, *evasive* functions are the class of functions such that it is hard to find an accepting input, given only black-box access to a functions. Note that compute-and-compare functions are evasive.

use of a natural but weaker notion of correctness *with respect to a distribution*. We note that general SSL was shown to be impossible [5], and that [1] mentions how *learnable* functions cannot be copy-protected. It is thus natural that we focus our efforts on achieving SSL for compute-and-compare circuits, which is a family of functions that is not learnable.

In more detail, we follow the security notion of [11], which postulates a game between a challenger, and a pirate Pete. Upon sampling a circuit from a given distribution, the challenger encodes the circuit and sends it to Pete. Pete then produces a register that he returns to the challenger who performs a *verification*; upon successful verification, we continue the game (otherwise, we abort), by presenting to Pete a challenge input $x \in \{0,1\}^n$ (chosen according to a given distribution). The scheme is ϵ-*secure* if we can bound the probability that Pete correctly evaluates the circuit on the challenge input x, to be within ϵ of his trivial guessing probability. Here, trivially guessing means that Pete answers the challenge by seeing only x, i.e., disregarding all other information obtained by interacting with the challenger. Thus, security is defined relative to the distribution on the circuits and on the challenges. For SSL, η-correctness is defined with respect to an input distribution, and means that, up to some error term η, the honest evaluation on an encoded circuit produces the correct outcome, *in expectation*.[8]

We show how to achieve SSL with respect to the uniform distribution on point functions, and the challenge distribution which samples uniformly from the distribution where the correct response is 0 or 1 (with equal probability) — denoted $T_p^{(1/2)}$. Our technique is a reduction from SSL to *honest-malicious* copy protection, as well as a new construction for quantum honest-malicious copy protection (with respect to essentially the same distributions as stated above). Prior work noted, informally, that copy protection implies SSL [5]. Here, we formally show that our new and weaker (and thus easier-to-achieve) notion of copy protection (see below) implies SSL. Our work focuses on achieving SSL for point functions; by applying our result with [11] this implies SSL for compute-and-compare circuits.

Honest-Malicious Copy Protection. We define a new security model for copy protection: *honest-malicious* copy protection.[9] Here, we consider a game between a challenger, a pirate (Pete), and two evaluators. Importantly, the first evaluator, Bob, is *honest* (meaning that he will execute the legitimate evaluation procedure) and the second evaluator, Charlie, is *malicious*. In copy protection, we want to bound the probability that, after each receiving a quantum register from Pete,

[8] This notion is weaker than the more common notion of correctness that holds for *all* inputs. However, in Section 4, we give evidence that achieving this stronger notion of correctness may be possible, by showing that for the standard notion of copy protection (against two malicious evaluators), correctness in expectation implies worst-case correctness, which would then imply worst-case correctness for SSL.

[9] This is a stronger notion of security than *infinite term SSL* as defined in [5], which is a form of copy protection where both evaluators are honest, and is achieved in [5] under strong assumptions.

who takes as input a single copy-protected program, the two evaluators (who cannot communicate), are *both* able to correctly evaluate the encoded circuit. Following [11], this is formalized by a game, parameterized by a distribution on the input circuits, and a corresponding challenge distribution on pairs of n-bit strings. A challenger samples a circuit, encodes it using the copy protection scheme and sends the encoding to Pete who creates the two registers. Then a challenge pair (x_1, x_2) is sampled from the challenge distribution; Bob receives x_1 while Charlie receives x_2. They *win* if they each produce the correct output of the original circuit evaluated on x_1 and x_2, respectively. An honest-malicious copy protection scheme is ϵ-*secure* for the given distributions if the probability that the evaluators win the game is within ϵ of the success probability of the trivial strategy that is achievable when Bob gets the full encoding and Charlie guesses to the best of his ability without interacting with Pete. As in the case of SSL, η-correctness for copy protection is defined with respect to an input distribution, and means that, up to some constant η, the honest evaluation on an encoded circuit produces the correct outcome, *in expectation*.[10]

We establish the relevance of honest-malicious copy protection by showing that, for general functions, honest-malicious copy protection implies SSL.

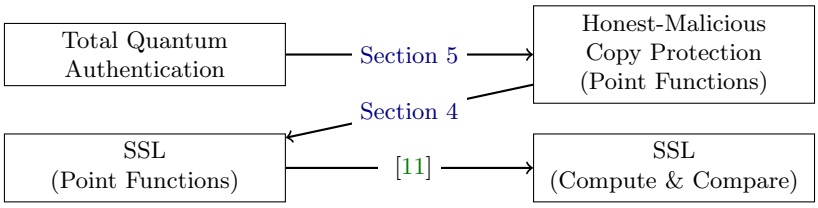

Fig. 1. Relations between various notions considered in this work.

In order to complete our main result, we show how to achieve honest-malicious copy protection for point functions, where the challenge distribution is given by $(T_p^{(1/2)} \times T_p^{(1/2)})$, and correctness is also with respect to $T_p^{(1/2)}$. To the best of our knowledge, this is the first unconditional copy protection scheme; via the above reduction, it yields the first SSL scheme without assumptions. See Figure 1 for a pictorial representation of the sequence of results. Our idea is to use a generic *quantum message authentication scheme (QAS)* that satisfies the *total authentication* property [17]. Briefly, a QAS is a private-key scheme with an encoding and decoding procedure such that the probability that the decoding accepts *and* the output of the decoding in *not* the original message is small. Security of a *total* QAS is defined in terms of the existence of a *simulator* that reproduces the auxiliary register that an adversary has after attacking an encoded system, *whenever* the verification accepts. An important feature of a total QAS is that essentially no information about the key is leaked if the client accepts the authentication.

[10] See Footnote 8.

The main insight for the construction of honest-malicious copy protection for point functions from a total QAS is to associate the key to the QAS with the point p in the point function. A copy-protected program is thus an encoding of an arbitrary (but fixed) state $|\psi\rangle$ into a total QAS, using p as the key. Given p', the evaluation of the point function encoding is the QAS verification *with the key* p'. We thus get correctness in the case $p' = p$ from the correctness of the QAS; correctness in expectation for $p' \neq p$ follows with a bit more work. Importantly, the *total* security property of the QAS gives us a handle on the auxiliary register that an adversary holds, *in the case that the verification accepts*. Since Bob is honest, his evaluation corresponds to the QAS verification map; in the case that Bob gets the challenge $x_1 = p$, we use the properties of the total QAS to reason about Charlie's register, and we are able to show that Charlie's register cannot have much of a dependence on p, which is to say that Charlie's outcome is necessarily independent of p. This is sufficient to conclude that Bob and Charlie cannot win the copy protection game for a uniform point with probability much better than the trivial strategy in which Charlie makes an educated guess, given the challenge x_2. We note that total authentication is known to be satisfied by a scheme based on 2-designs [4], as well as by the *strong trap code* [14]. Putting all of the above together, we obtain our main result, which is an explicit SSL scheme for point functions $P_p : \{0,1\}^n \to \{0,1\}$ which is $O(2^{-\frac{n}{16}})$-correct (on average) and $O(2^{-\frac{n}{32}})$-secure, under uniform sampling of p and where the challenge distribution is $T_p^{(1/2)}$.[11] We note the similarity between our approach for achieving honest-malicious copy protection and the approach in [18] in achieving tamper-evident encryption, based on quantum authentication codes. We also mention a similarity with the blueprint in [11], which also produces a copy-protected program starting from a private-key encryption scheme (in this case, the one of [8]), associates a point with the key, and uses a type of verification of the integrity of the plaintext after decryption as the evaluation method.

Too good to be true? We emphasize that our results require no assumptions at all, which is to say that the result is in the standard model (as opposed to, say the QROM), and does not rely on any assumption on the computational power of the adversary. That either copy protection or SSL should be achievable in this model is very counter-intuitive, hence we explain here how we circumvent related impossibility results. In short, our work strikes a delicate balance between correctness and security, in order to achieve the best of both worlds.

Prior work [1] defines quantum copy protection assuming the adversary is given *multiple identical* copies of the same copy-protected state. Under this model, it is possible to show how an unbounded adversary can distinguish between the copy-protected programs for different functions [1], which makes unconditionally secure copy protection impossible. In our scenario, we allow only a *single* copy of the program state, hence this reasoning is not applicable.

[11] This is achieved by instantiating the copy protection scheme from Section 5 with a total quantum authentication scheme given by Lemma 3 and using it in the SSL construction of Section 4.3.

Next, consider a scheme (either copy protection or SSL) that is *perfectly correct*, meaning that the outcome of the evaluation procedure is a deterministic bit. Clearly, such a scheme cannot be secure against unbounded adversaries, since *in principle*, there is a sequence of measurements that an unbounded adversary can perform (via purification and rewinding), in order to perfectly obtain the truth table of the function. We conclude that perfectly correct schemes cannot satisfy our notion of unconditional security for copy protection.

We note that our scheme is, by design, not perfectly correct. This can be seen by reasoning about the properties of the QAS: in any QAS, it is necessary that, for a fixed encoding with key k, there are a number of keys on which the verification accepts. The reason why this is true is similar to the argument above regarding perfect correctness: if this were not true, then the QAS (which is defined with respect to unbounded adversaries) would not be secure, since an adversary could in principle find k by trying all keys (coherently, so as to not disturb the quantum state) until one accepts. Somewhat paradoxically, it is this imperfection in the correctness that thus allows the unconditional security. Another way to understand the situation is that the honest evaluation in our copy protection (or SSL) scheme will unavoidably slightly damage the quantum encoding (even if performed coherently). In a brute-force attack, these errors necessarily accumulate to the point of rendering the program useless, and therefore the brute-force attack fails.

1.2 Open Problems

Our work leaves open a number of interesting avenues. For instance: (i) Could we show the more standard notion of correctness of our scheme, that is, correctness with respect to *any* distribution? (ii) Is unconditional SSL achievable for a richer class of functions? (iii) Can our results on copy protection be extended to hold against *two* malicious evaluators? In Section 4, we show that (i) and (iii) are related, by establishing that a point function copy protection scheme that is secure against two malicious evaluators and satisfies average correctness can be turned into a scheme that also satisfies the more standard notion of correctness.

1.3 Outline

The remainder of this document is structured as follows. In Section 2, we give background information on notation, basic notions and quantum message authentication. In Section 3, we define correctness and security for quantum copy protection and SSL. In Section 4, we show the connection between malicious-malicious security and standard correctness, as well as links between honest-malicious copy protection and SSL. Finally, our main technical construction of honest-malicious copy protection from any total QAS is given in Section 5. Note that some technical details can be found in the full version[12].

[12] The full version is available at: `arXiv:2101.12739`.

2 Preliminaries

2.1 Notation

All Hilbert spaces in this work are complex and of finite dimension. We will usually denote a Hilbert space using a sans-serif font such as S or H. We will often omit the tensor symbol when taking the tensor product of two Hilbert spaces, i.e.: $\mathsf{A} \otimes \mathsf{B} = \mathsf{AB}$. We use the Dirac notation throughout, which is to say that $|\psi\rangle \in \mathsf{H}$ denotes a vector and $\langle\psi| : \mathsf{H} \to \mathbb{C}$ denotes the corresponding linear map in the dual space. Finally, Hilbert spaces may be referred to as "registers", acknowledging that they sometimes model physical objects which may be sent, kept, discarded, etc., by participants in quantum information processing tasks.

The set of linear operators, unitary operators, and density operators on a Hilbert space H are denoted by $\mathcal{L}(\mathsf{H})$, $\mathcal{U}(\mathsf{H})$, and $\mathcal{D}(\mathsf{H})$ respectively. A linear operator may be accompanied by a subscript indicating the Hilbert space on which it acts. This will be useful for bookkeeping and to omit superfluous identities. For example, if $L_{\mathsf{A}} \in \mathcal{L}(\mathsf{A})$ and $|\psi\rangle_{\mathsf{AB}} \in \mathsf{AB}$, then $L_{\mathsf{A}} |\psi\rangle_{\mathsf{AB}} = (L_{\mathsf{A}} \otimes I_{\mathsf{B}}) |\psi\rangle_{\mathsf{AB}}$.

We recall [23] that the trace norm, or Schatten 1-norm, of a linear operator is given by $\|X\|_1 = \max_{U \in \mathcal{U}(\mathsf{A})} |\langle U|X\rangle|$ where $\langle U|X\rangle = \mathrm{Tr}\left[U^\dagger X\right]$. The trace distance between two linear operators is then given by $\Delta(X,Y) = \frac{1}{2}\|X - Y\|_1$. If $\Delta(X,Y) \leq \epsilon$, we write $X \approx_\epsilon Y$. We also give a technical lemma pertaining to the trace distance between bipartite states of a particular form. For completeness, a proof is given in the full version.

Lemma 1. *Let A and B be Hilbert spaces and $\{|\psi_j\rangle\}_{j \in J} \subseteq \mathsf{A}$ be a set of orthonormal vectors. Then, for any sets of linear operators $\{X_j\}_{j \in J}$ and $\{Y_j\}_{j \in J}$ on B, we have that*

$$\Delta\left(\sum_{j \in J} |\psi_j\rangle\langle\psi_j| \otimes X_j, \sum_{j \in J} |\psi_j\rangle\langle\psi_j| \otimes Y_j\right) = \sum_{j \in J} \Delta\left(X_j, Y_j\right). \tag{1}$$

For a distribution D on a set S, we will use the notation $x \leftarrow D$ to denote that variable x is sampled from D, and $D(x)$ to denote the probability that a given $x \in S$ is sampled. If S is finite, for a given map $f : S \to \mathbb{C}$ we write

$$\mathop{\mathbb{E}}_{x \leftarrow D} f(x) = \sum_{x \in S} D(x) f(x). \tag{2}$$

If no distribution is specified or clear from context, we assume a uniform distribution, which is to say that $\mathbb{E}_x f(x) = |S|^{-1} \cdot \sum_{x \in S} f(x)$.

Two classes of functions will be of particular interest in this work: point functions and compute-and-compare functions. For any $p \in \{0,1\}^n$, the map $P_p : \{0,1\}^n \to \{0,1\}$ satisfying $P_p(x) = 1 \iff x = p$ is called the point function for p. For any function $f : \{0,1\}^n \to \{0,1\}^m$ and bit string $y \in \{0,1\}^m$, the map $\mathsf{CC}^f_y : \{0,1\}^n \to \{0,1\}$ which satisfies $\mathsf{CC}^f_y(x) = 1 \iff f(x) = y$ is the compute-and-compare function of f and y. We note that point functions can be seen as compute-and-compare functions where f is the identity.

An n-bit Boolean circuit is a circuit taking as input an element of $\{0,1\}^n$ and producing as output a single bit. Throughout this work, we will denote a family of Boolean circuits on n bits as $\mathcal{C}$.

2.2 Quantum Authentication

We recall the definition of a total quantum authentication scheme [17] and highlight a few properties of such schemes.

Definition 1. *An authentication scheme* $\mathsf{QAS} = \{(\mathsf{QAS.Auth}_k, \mathsf{QAS.Ver}_k)\}_{k\in\mathcal{K}}$ *for the Hilbert space* M *is a pair of keyed CPTP maps*

$$\mathsf{QAS.Auth}_k : \mathcal{L}(\mathsf{M}) \to \mathcal{L}(\mathsf{Y}) \quad \textit{and} \quad \mathsf{QAS.Ver}_k : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{MF}) \tag{3}$$

and where F *admits* $\{|Acc\rangle, |Rej\rangle\}$ *as an orthonormal basis. Moreover, these maps are such that for all states* $\rho \in \mathcal{D}(\mathsf{M})$ *and all keys* $k \in \mathcal{K}$ *we have that*

$$\mathsf{QAS.Ver}_k \circ \mathsf{QAS.Auth}_k(\rho) = \rho \otimes |Acc\rangle\langle Acc| . \tag{4}$$

We assume throughout this work that the keys for an authentication scheme are generated uniformly at random.

To facilitate our analysis, we will make the same simplifying assumptions as in [17] on any quantum authentication scheme considered in this work.

1. We assume that $\mathsf{QAS.Auth}_k$ can be modeled by an isometry. Specifically, we assume that
$$\mathsf{QAS.Auth}_k(\rho) = A_k \rho A_k^\dagger \tag{5}$$
for some isometry $A_k \in \mathcal{L}(\mathsf{M}, \mathsf{Y})$.
2. For all keys $k \in \mathcal{K}$, as A_k is an isometry, $A_k A_k^\dagger$ is the projector onto the image of A_k. In other words, it projects onto valid authenticated states for the key k. We then assume that $\mathsf{QAS.Ver}_k$ is given by the map
$$\rho \mapsto A_k^\dagger \rho A_k \otimes |\mathrm{Acc}\rangle\langle\mathrm{Acc}| + \mathrm{Tr}\left[\left(I - A_k A_k^\dagger\right)\rho\right] \cdot \frac{I}{\dim(\mathsf{M})} \otimes |\mathrm{Rej}\rangle\langle\mathrm{Rej}| . \tag{6}$$
In other words, $\mathsf{QAS.Ver}_k$ verifies if the state is a valid encoded state. If it is, then it inverts the authentication procedure and adds an "accept" flag. If it is not, then it outputs the maximally mixed state and adds a "reject" flag.

Finally, we will also define the map $\mathsf{QAS.Ver}'_k : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{M})$ by

$$\rho \mapsto (I_\mathsf{M} \otimes \langle \mathrm{Acc}|_\mathsf{F})\, \mathsf{QAS.Ver}_k(\rho)\, (I_\mathsf{M} \otimes |\mathrm{Acc}\rangle_\mathsf{F}) = A_k^\dagger \rho A_k. \tag{7}$$

Essentially, this map outputs a subnormalized state corresponding to the state of the message register M conditioned on the verification procedure accepting the state. In particular, note that the probability that the verification procedure accepts the state ρ when using the key k is given by $\mathrm{Tr}\left(\mathsf{QAS.Ver}'_k(\rho)\right)$.

Definition 1 does not make any type of security guarantee on an authentication scheme. It only specifies a syntax, Eq. (3), and a correctness guarantee, Eq. (4). The following definition describes the security guarantee of an ϵ-total quantum authentication scheme. Note that this security definition differs from some early notions of security for quantum authentication schemes [6, 15].

Definition 2. *An authentication scheme QAS is an ϵ-total authentication scheme if for all CPTP maps $\Phi : \mathcal{L}(\mathsf{YZ}) \to \mathcal{L}(\mathsf{YZ})$ there exists a completely positive trace non-increasing map $\Psi : \mathcal{L}(\mathsf{Z}) \to \mathcal{L}(\mathsf{Z})$ such that*

$$\mathop{\mathbb{E}}_{k\in\mathcal{K}} |k\rangle\langle k| \otimes \mathsf{QAS.Ver}'_k \circ \Phi \circ \mathsf{QAS.Auth}_k(\rho) \approx_\epsilon \mathop{\mathbb{E}}_{k\in\mathcal{K}} |k\rangle\langle k| \otimes \mathsf{QAS.Ver}'_k \circ \Psi \circ \mathsf{QAS.Auth}_k(\rho) \tag{8}$$

for any state $\rho \in \mathcal{D}(\mathsf{MZ})$.

A key difference between the "total" security definition given in [17] and previous security definitions for authentication schemes is the explicit $|k\rangle\langle k|$ state which appears in Eq. (8). This key register will be used, with the help of Lemma 1, in some of our technical arguments, such as the proof of Lemma 5.

Note that our discussion, unlike the one in [17], omits adding another register S to model all other information that a sender and receiver could share as part of a larger protocol but which is not directly implicated in the authentication scheme. Such a register is not needed in our analysis.

Next, we give a lemma which upper bounds the probability that any fixed state is accepted by the verification procedure, when averaged over all possible keys. This allows us to make statements on what happens if an authenticated state is verified with the wrong key—a scenario which is not usually considered for authentication schemes. Intuitively, no quantum authentication scheme can admit such a fixed state ρ that is accepted with high probability over all keys, since otherwise an adversary could insert such a ρ in place of any authenticated message, and this modification would go undetected with high probability. The formal proof of Lemma 2 is given in the full version.

Lemma 2. *Let QAS be an ϵ-total authentication scheme on the Hilbert space M of dimension greater or equal to 2. Then, for any $\rho \in \mathcal{D}(\mathsf{Y})$, we have that*

$$\mathop{\mathbb{E}}_{k\in\mathcal{K}} \mathrm{Tr}\left[\mathsf{QAS.Ver}'_k(\rho)\right] \leq 2\epsilon. \tag{9}$$

Finally, we give an existence lemma for total quantum authentication schemes satisfying certain parameters (Lemma 3). The proof is given in the full version. It essentially follows from a theorem describing how unitary 2-designs (as introduced in [12]) can be used to construct total quantum authentication schemes [4] and then choosing a suitable unitary 2-design [10]. A few additional technical arguments are needed to ensure that the key set is precisely the bit strings of a given length.

Lemma 3. *For any strictly positive integers n and k, there exists a $\left(5 \cdot 2^{\frac{5n-k}{16}}\right)$-total quantum authentication scheme on n qubits with key set $\{0,1\}^k$.*

3 Definitions

Here, we define quantum copy protection (Section 3.1) and secure software leasing (Section 3.2), along with their correctness and security notions. All of our definitions are for Boolean circuits only, where the input is a binary string, and the output is a single bit. Finally, we define distributions on circuits and inputs which will often be used in this work in Section 3.3.

3.1 Quantum Copy Protection

We present our definition of a copy protection scheme, following the general lines of [11]. We first define the functionality (Definition 3) and correctness (Definition 4) of a copy protection scheme. We then define honest-malicious security in Definition 6, which can be contrasted with the usual definition of security (which we call malicious-malicious security) given in Definition 7. We note that we have rephrased the definition in [11] in terms of the more standard cryptographic notion where the parameter in the definition (here, we use ϵ) characterizes the *insecurity* of a game (and hence, we strive for schemes where ϵ is small).

First, we define the functionality of *quantum copy protection*.

Definition 3 (Quantum copy protection scheme). *Let $\mathcal{C}$ be a set of n-bit Boolean circuits. A* quantum copy protection *scheme for $\mathcal{C}$ is a pair of quantum circuits $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ such that for some space Y:*

1. *$\mathsf{CP.Protect}(C)$: takes as input a Boolean circuit $C \in \mathcal{C}$, and outputs a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$.*
2. *$\mathsf{CP.Eval}(\rho, x)$: takes a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$ and string $x \in \{0,1\}^n$ as inputs and outputs a bit b.*

We will interpret the output of $\mathsf{CP.Protect}$ and $\mathsf{CP.Eval}$ as quantum states on Y and $\mathbb{C}^2$, respectively, so that, for example, for any bit b, string x and program ρ, $\mathrm{Tr}[|b\rangle\langle b| \, \mathsf{CP.Eval}(\rho, x)]$ is the probability that $\mathsf{CP.Eval}(\rho, x)$ outputs b.

Definition 4 (η-Correctness of copy protection). *A* quantum copy protection *scheme for a set of n-bit circuits $\mathcal{C}$, CP, is* η-correct *with respect to a family of distributions on n-bit strings $\{T_C\}_{C\in\mathcal{C}}$, if for any $C \in \mathcal{C}$ and $\rho = \mathsf{CP.Protect}(C)$, the scheme satisfies*

$$\mathop{\mathbb{E}}_{x\leftarrow T_C} \mathrm{Tr}[|C(x)\rangle\langle C(x)| \, \mathsf{CP.Eval}(\rho, x)] \geq 1 - \eta. \tag{10}$$

Our notion of correctness differs from that of [11] and other previous work by being defined with respect to a family of distributions (see Section 1.2). However, if the scheme is η-correct with respect to all families of distributions, then we recover the more standard definition of correctness.

We now define the notion of security for a copy protection scheme against an adversary $\mathcal{A} = (\mathcal{P}, \mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{P}$ (Pete) is the *pirate*, and $\mathcal{A}_1$ (Bob) and $\mathcal{A}_2$

(Charlie) are *users* (see Fig. 2). We use the PiratingGame from [11] as the basis of our security game between a challenger and $\mathcal{A}$. The game is parametrized by: (i) a distribution D on the set of circuits $\mathcal{C}$, and (ii) a set of distributions $\{D_C\}_{C\in\mathcal{C}}$ over pairs of input strings in $\{0,1\}^n \times \{0,1\}^n$, called the *challenge distributions*.

The CP game $\mathsf{PiratingGame}_{\mathcal{A},\mathsf{CP}}$

1. The challenger samples $C \leftarrow D$ and sends $\rho = \mathsf{CP.Protect}(C)$ to $\mathcal{P}$.
2. $\mathcal{P}$ outputs a state σ on registers $\mathsf{A}_1, \mathsf{A}_2$ and sends A_1 to $\mathcal{A}_1$ and A_2 to $\mathcal{A}_2$.
3. At this point, $\mathcal{A}_1$ and $\mathcal{A}_2$ are separated and cannot communicate. The challenger samples $(x_1, x_2) \leftarrow D_C$ and sends x_1 to $\mathcal{A}_1$ and x_2 to $\mathcal{A}_2$.
4. $\mathcal{A}_1$ returns a bit b_1 to the challenger and $\mathcal{A}_2$ returns a bit b_2.
5. The challenger outputs 1 if and only if $b_1 = C(x_1)$ and $b_2 = C(x_2)$, in which case, we say that $\mathcal{A}$ wins the game.

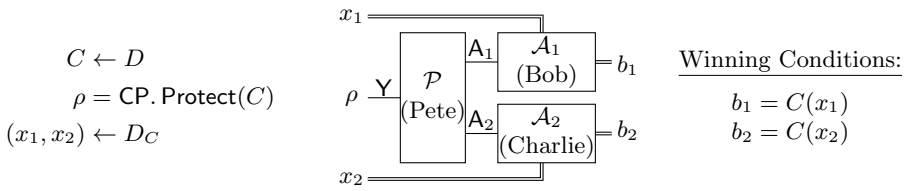

Fig. 2. The pirating game $\mathsf{PiratingGame}_{\mathcal{A},\mathsf{CP}}$

In previous work on copy protection, the adversary is assumed to control $\mathcal{P}$, $\mathcal{A}_1$ and $\mathcal{A}_2$, whose behaviour can be arbitrary (or, in some cases, computationally bounded). This models a setting where the potential users of pirated software are aware that the software is pirated, and willing to run their software in some non-standard way in order to make use of it. We refer to this setting as the *malicious-malicious* setting. In this setting, the action of the adversary $\mathcal{A} = (\mathcal{P}, \mathcal{A}_1, \mathcal{A}_2)$ can be specified by:

1. an arbitrary CPTP map $\Phi_{\mathcal{P}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{A}_1\mathsf{A}_2)$, representing the action of $\mathcal{P}$, where A_1 and A_2 are arbitrary spaces;
2. arbitrary two-outcome projective measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A_1, such that $\mathcal{A}_1$ (Bob) performs the measurement $\{\Pi_{x_1}, I - \Pi_{x_1}\}$ on input x_1 to obtain his output bit b_1; and
3. arbitrary two-outcome projective measurements $\{\Pi'_x\}_{x\in\{0,1\}^n}$ on A_2, such that $\mathcal{A}_2$ (Charlie) performs the measurement $\{\Pi'_{x_2}, I - \Pi'_{x_2}\}$ on input x_2 to obtain his output bit b_2.

Note that we restrict our attention to projective measurements for $\mathcal{A}_1$ and $\mathcal{A}_2$. Indeed, by a purification argument, any strategy using non-projective measure-

ments is equivalent to a strategy with projective measurements and the extra auxillary states needed by $\mathcal{A}_1$ and $\mathcal{A}_2$ can be provided by $\mathcal{P}$.

In contrast, one could also imagine a scenario in which users are honest, and will therefore try to evalute the program they receive from $\mathcal{P}$ by running $\mathsf{CP.Eval}$. In that case, while $\mathcal{P}$ can still perform an arbitrary CPTP map, $\mathcal{A}_1$ and $\mathcal{A}_2$ are constrained to run $\mathsf{CP.Eval}$. It is potentially easier to design copy protection in this weaker setting, which we call the *honest-honest* setting, since the adversary is more constrained. We will consider an intermediate setting.

Diverging from previous work, we will focus on a special type of adversary, where $\mathcal{A}_1$ (Bob) performs the *honest* evaluation procedure, while $\mathcal{A}_2$ (Charlie) performs an arbitrary measurement. (See Section 1.1 for a discussion of this model). Specifically, we consider the following type of adversary.

Definition 5. *An* honest-malicious adversary *for the pirating game is an adversary of the form* $\hat{\mathcal{A}} = (\mathcal{P}, \mathsf{CP.Eval}, \mathcal{A}_2)$*, where* $\mathcal{P}$ *implements an arbitrary CPTP map* $\Phi_{\mathcal{P}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{YA}_2)$*,* A_2 *is any space, and* $\mathcal{A}_2$ *is specified by a set of arbitrary two-outcome measurements* $\{\Pi_x\}_{x\in\{0,1\}^n}$ *on* A_2*.*

For a fixed scheme $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ for a set of n-bit circuits $\mathcal{C}$, we define *honest-malicious* security with respect to distributions D and $\{D_C\}_{C\in\mathcal{C}}$ in terms of the best possible winning probability, $\Pr\left[\mathsf{PiratingGame}_{\hat{\mathcal{A}},\mathsf{CP}}\right]$, over honest-malicious adversaries $\hat{\mathcal{A}}$. Observe that there is one strategy that $\mathcal{P}$ can always facilitate, which is to pass the intact program to Bob and then let Charlie locally produce his best guess of the output, based on prior knowledge of D and $\{D_C\}_{C\in\mathcal{C}}$[13]. This leads to a winning probability for the above game which is truly trivial to achieve, in the sense that Charlie is using a strategy that does not take any advantage of the interaction with the pirate $\mathcal{P}$. In fact, assuming the scheme is η-correct with respect to the distribution family $\{T_C\}_{C\in\mathcal{C}}$ where T_C is Bob's marginal of D_C, Bob will always produce the correct answer, except with probability η. Indeed, Charlie simply considers the most likely output, given his input, thereby upper bounding the winning probability with Charlie's maximum guessing probability[14].

Formally, we define $p^{\mathrm{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}}$ as follows. The distributions D and $\{D_C\}_{C\in\mathcal{C}}$ yield a joint distribution $\tilde{D}$ on $\mathcal{C}\times\{0,1\}^n$ by first sampling $C \leftarrow D$ and then sampling $(x_1, x_2) \leftarrow D_C$ and only taking the x_2 component. Let $\hat{D}$ be the marginal distribution of x_2 from $\tilde{D}$ and, for every x, let $\hat{D}_x$ be the marginal distribution of C from $\tilde{D}$, conditioned on $x_2 = x$. Then,

$$p^{\mathrm{marg}}_{D,\{D_C\}_{C\in\mathcal{C}}} = \mathop{\mathbb{E}}_{x\leftarrow\hat{D}} \max_{b\in\{0,1\}} \Pr_{C\leftarrow\hat{D}_x}[C(x) = b]. \tag{11}$$

[13] There are other trivial strategies, *e.g.*, where Charlie gets an intact program register and Bob does not, but this is a more restricted trivial strategy, since Bob is constrained to evaluate the program honestly.

[14] The winning probability may be less than this. By the union bound, even though Bob's and Charlie's inputs are not independent, the overall success probability will be at least $p^{\mathrm{marg}} - \eta$, and we will be considering situations where η is small.

This is different from the security notion in [11] where the trivial guessing probability is optimized over both users. For intuition, note that p^{marg} is always at least $1/2$, since Charlie can always output a random bit that is correct with probability $1/2$. Depending on the specific input and challenge distributions, it may be larger. We now state the main security notion for this work.

Definition 6 (Honest-malicious security). *A copy protection scheme* $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ *for a set of* n*-bit circuits* $\mathcal{C}$ *is* ϵ-honest-malicious secure with respect to the distribution D and challenge distributions $\{D_C\}_{C\in\mathcal{C}}$ *if for all honest-malicious adversaries* $\hat{\mathcal{A}}$*,*

$$\Pr\Big[\mathsf{PiratingGame}_{\hat{\mathcal{A}},\mathsf{CP}}\Big] \leq p^{marg} + \epsilon\,, \tag{12}$$

where $p^{marg} = p^{marg}_{D,\{D_C\}_{C\in\mathcal{C}}}$.

We re-iterate that our definition for honest-malicious security is *statistical*: it makes no assumption on the computational power of $\hat{\mathcal{A}}$ (see Section 1.1).

Finally, if we modify the above definition by allowing arbitrary adversaries $\mathcal{A} = (\mathcal{P}, \mathcal{A}_1, \mathcal{A}_2)$, and letting $\bar{p}^{\text{marg}}$ denote the optimal trivial guessing probability, as in Eq. (11) but over *both* adversaries (see also [11]), we recover the more standard security definition, which we call *malicious-malicious* security:

Definition 7 (Malicious-malicious security). *A copy protection scheme* CP *for a set of* n*-bit circuits* $\mathcal{C}$ *is* ϵ-malicious-malicious secure with respect to the distribution D and challenge distributions $\{D_C\}_{C\in\mathcal{C}}$ *if for* all *adversaries* $\mathcal{A}$*,*

$$\Pr\big[\mathsf{PiratingGame}_{\mathcal{A},\mathsf{CP}}\big] \leq \bar{p}^{marg} + \epsilon. \tag{13}$$

3.2 Secure Software Leasing

We define Secure Software Leasing (SSL) below. As with copy protection, the basic scheme and security game mirror [11] but we diverge from them in our exact notions of correctness and security. We first define the functionality (Definition 8) and correctness (Definition 9) of an SSL scheme, followed by its security (Definition 10).

Definition 8 (Secure software leasing (SSL)). *Let* $\mathcal{C}$ *be a set of* n*-bit Boolean circuits. A* secure software leasing *scheme for* $\mathcal{C}$ *is a tuple of quantum circuits* $\mathsf{SSL} = (\mathsf{SSL.Gen}, \mathsf{SSL.Lease}, \mathsf{SSL.Eval}, \mathsf{SSL.Verify})$ *such that for some space* Y*:*

1. $\mathsf{SSL.Gen}$*: outputs a secret key* sk*.*
2. $\mathsf{SSL.Lease}(\mathsf{sk}, C)$*: takes as input a secret key* sk *and a circuit* $C \in \mathcal{C}$*, and outputs a quantum state* $\rho \in \mathcal{D}(\mathsf{Y})$*.*
3. $\mathsf{SSL.Eval}(\rho, x)$*: takes as input a quantum state* $\rho \in \mathcal{D}(\mathsf{Y})$ *and input string* $x \in \{0,1\}^n$ *and outputs a bit* b*.*
4. $\mathsf{SSL.Verify}(\mathsf{sk}, \sigma, C)$*: takes a secret key* sk*, a circuit* $C \in \mathcal{C}$ *and a quantum state* $\sigma \in \mathcal{D}(\mathsf{Y})$*, and outputs a bit* v *indicating acceptance or rejection.*

Definition 9 (η-Correctness of SSL). *A* secure software leasing *scheme for $\mathcal{C}$, SSL, is* η-correct *with respect to a family of distributions on n-bit strings $\{T_C\}_{C\in\mathcal{C}}$, if for any $C \in \mathcal{C}$, $\mathsf{sk} \leftarrow \mathsf{SSL.Gen}$, and $\rho = \mathsf{SSL.Lease}(\mathsf{sk}, C)$, the scheme satisfies:*

- *Correctness of Evaluation:* $\mathop{\mathbb{E}}_{x\leftarrow T_C} \mathrm{Tr}\left(|C(x)\rangle\langle C(x)|\, \mathsf{SSL.Eval}(\rho, x)]\right) \geq 1 - \eta$,
- *and Correctness of Verification:* $\mathrm{Tr}\left(|1\rangle\langle 1|\, \mathsf{SSL.Verify}(\mathsf{sk}, \rho, C)\right) \geq 1 - \eta$.

In the above definition, recall that for $b \in \{0, 1\}$, $\mathrm{Tr}\left(|b\rangle\langle b|\, \mathsf{SSL.Verify}(\mathsf{sk}, \rho, C)\right)$ is the probability that $\mathsf{SSL.Verify}(\mathsf{sk}, \rho, C)$ outputs the bit b, and similarly for $\mathrm{Tr}\left(|b\rangle\langle b|\, \mathsf{SSL.Eval}(\rho, x)\right)$.

When a scheme SSL is η-correct with respect to every distribution, we recover the more standard notion of correctness.

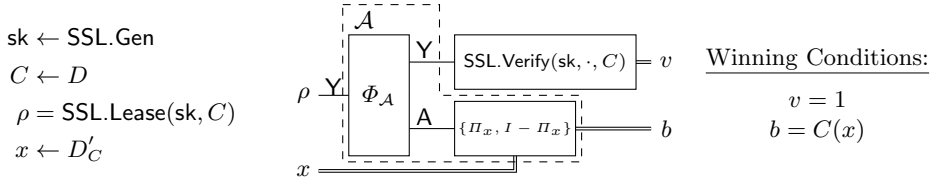

Fig. 3. The SSL game $\mathsf{SSLGame}_{\mathcal{A},\mathsf{SSL}}$, where the behaviour of $\mathcal{A}$ is specified by a CPTP map $\Phi_{\mathcal{A}}$ and a set of two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$.

We base our security game, between a challenger (in this case a *Lessor*) and an adversary $\mathcal{A}$, on the $\mathsf{SSLGame}$ from [11]. The game is parametrized by a distribution D over circuits in $\mathcal{C}$, and a set of challenge distributions $\{D'_C\}_{C\in\mathcal{C}}$ over inputs $\{0, 1\}^n$.

The SSL game $\mathsf{SSLGame}_{\mathcal{A},\mathsf{SSL}}$

1. The Lessor samples $C \leftarrow D$ and runs $\mathsf{SSL.Gen}$ to obtain a secret key sk. She then sends $\rho = \mathsf{SSL.Lease}(\mathsf{sk}, C)$ to $\mathcal{A}$.
2. $\mathcal{A}$ produces a state σ on registers YA and sends register Y back to the Lessor and keeps A.
3. (*Verification phase.*) The Lessor runs $\mathsf{SSL.Verify}$ on Y, the circuit C and the secret key sk and outputs the resulting bit v. If $\mathsf{SSL.Verify}$ accepts ($v = 1$), the game continues, otherwise it aborts and $\mathcal{A}$ loses.
4. The Lessor samples an input $x \leftarrow D'_C$ and sends x to $\mathcal{A}$.
5. $\mathcal{A}$ returns a bit b to the Lessor.
6. The Lessor outputs 1 if and only if $b = C(x)$ and $v = 1$, in which case, we say $\mathcal{A}$ "wins" the game.

An adversary $\mathcal{A}$ for $\mathsf{SSLGame}$ can be described by: an arbitrary CPTP map $\Phi_{\mathcal{A}} : \mathcal{L}(\mathsf{Y}) \to \mathcal{L}(\mathsf{YA})$ for some arbitrary space A, representing the action of $\mathcal{A}$

in Step 2; and a set of two-outcome measurements $\{\Pi_x\}_{x\in\{0,1\}^n}$ on A such that given challenge x in Step 4, $\mathcal{A}$ obtains the bit b in Step 5 by measuring A with $\{\Pi_x, I-\Pi_x\}$ (see Fig. 3).

As in Section 3.1, we define security with respect to the trivial strategy where $\mathcal{A}$ returns the program ρ to the Lessor in Step 2, and tries to guess the most likely value for b, given input x.

Formally, we define $p^{\text{triv}}_{D,\{D_C\}_{C\in\mathcal{C}}}$ as follows. The distributions D and $\{D_C\}_{C\in\mathcal{C}}$ yield a joint distribution $\tilde{D}$ on $\mathcal{C}\times\{0,1\}^n$ by first sampling $C\leftarrow D$ and then sampling $x\leftarrow D_C$. Let $\hat{D}$ be the marginal distribution of x from $\tilde{D}$ and, for every x', let $\hat{D}_{x'}$ be the marginal distribution of C from $\tilde{D}$, conditioned on $x=x'$. Then,

$$p^{\text{triv}}_{D,\{D'_C\}_{C\in\mathcal{C}}} = \mathop{\mathbb{E}}_{x\leftarrow\hat{D}} \max_{b\in\{0,1\}} \Pr_{C\leftarrow\hat{D}_x}[C(x)=b]. \tag{14}$$

The above equation is very similar to p^{marg} given in Equation (11). However, we point out that they are defined and used in different contexts. Specifically, in $\mathsf{PiratingGame}$ there are two parties, Bob and Charlie, who must be challenged with inputs on which to evaluate the function. However, there is only a single party attempting to evaluate the function at the end of $\mathsf{SSLGame}$. Thus, p^{marg} is defined with respect to the marginal distribution on Charlie's challenge generated by the joint challenge distribution. On the other hand, p^{triv} can be directly defined with respect to the single challenge issued in $\mathsf{SSLGame}$.

We now define the security of SSL as follows.

Definition 10 (Security of SSL). *An SSL scheme SSL for a set of n-bit circuits $\mathcal{C}$ is* ϵ-secure with respect to the distribution D and challenge distributions $\{D'_C\}_{C\in\mathcal{C}}$ *if for all adversaries $\mathcal{A}$,*

$$\Pr[\mathsf{SSLGame}_{\mathcal{A}}] \leq p^{triv} + \epsilon\,, \tag{15}$$

where $p^{triv} = p^{triv}_{D,\{D'_C\}_{C\in\mathcal{C}}}$.

Observe that, as in the case with Definition 6, our definition provides statistical guarantees for security as we impose no conditions on the adversaries.

3.3 Distributions for Point Functions

The definitions of correctness and security for copy protection and secure software leasing presented earlier in this section are parametrized by various distributions on the circuits that are encoded and the challenges that are issued.

In this section, we define notation for the distributions we will consider in the setting of point functions. First, we will consider security in the setting when the point function is chosen uniformly at random.

Definition 11. *We let R be the uniform distribution on the set of point functions $\{P_p \;:\; p\in\{0,1\}^n\}$. For simplicity, we will also use R to simply refer to the uniform distribution on $\{0,1\}^n$, as we often conflate a point p with its corresponding point function P_p.*

For a fixed point p, we will consider the distribution of inputs where p is sampled with probability $1/2$, and otherwise, a uniform $x \neq p$ is sampled.

Definition 12. *For any bit string $p \in \{0,1\}^n$, we define $T_p^{(1/2)}$ to be the distribution on $\{0,1\}^n$ such that*

- *p is sampled with probability $\frac{1}{2}$ and*
- *any $x \neq p$ is sampled with probability $\frac{1}{2} \cdot \frac{1}{2^n-1}$.*

This is a natural distribution in the setting of point functions, since it means that the function evaluates to a uniform random bit. This ensures that the output is non-trivial to guess — an adversary's advantage against challenge distributions of this form can be quantified by comparing it with their probability of correctly guessing a random bit. Furthermore, η-correctness with respect to this distribution, for some small η, ensures that evaluating the point is correct except with small probability, and that all but a small fraction of the other inputs are evaluated correctly except with small probability.

4 Generic Results on Definitions

Here, we give some generic results concerning definitions given in Section 3.

We first discuss the reusability of program states generated by copy protection or SSL schemes in Section 4.1. In Section 4.2, we outline how a copy protection scheme satisfying malicious-malicious security and average correctness can be used to obtain a scheme satisfying malicious-malicious security and the more standard definition of correctness. Next, in Section 4.3, we describe how an honest-malicious copy protection scheme for any set of circuits $\mathcal{C}$ can be turned into an SSL scheme for $\mathcal{C}$ (Theorem 2). In particular, this means that the copy protection scheme for point functions presented in Section 5 implies an SSL scheme for point functions. Finally, in Section 4.4, we refine a result from [11] (tailoring it to our definitions), to show that an SSL scheme for point functions can be used to construct an SSL scheme for compute-and-compare programs (Theorem 3).

4.1 Reusability of the Program

For ease of notation, we define both $\mathsf{CP}.\mathsf{Eval}(\rho, x)$ and $\mathsf{SSL}.\mathsf{Eval}(\rho, x)$ to take a quantum state $\rho \in \mathcal{D}(\mathsf{Y})$ and a string $x \in \{0,1\}^n$ as inputs and output a bit b. This can be extended [5] to a *reusable* scheme in a straightforward way, so that the evaluation procedure outputs a bit b together with a post-evaluated state $\tilde{\rho}$, which approximates ρ. In more details, we purify the evaluation procedure, and copy the output bit, before then undoing the evaluation procedure.

We claim that the above procedure, used to sequentially evaluate n inputs sampled from the same distribution with respect to which the scheme is η-correct, will produce all the n correct answers with a probability of at least $(1-n\sqrt{\eta})(1-4n\sqrt{\eta})$. We highlight that this bound is sufficient to show that, in an

asymptotic regime, a program state with negligible errors evaluated polynomially many times on randomly sampled inputs will give all the correct values with overwhelming probability.

We give the above probabilistic statement because it is impossible to give a precise figure for the number of times a program can be evaluated before it stops working altogether. Indeed, in some cases the evaluation of the program state could leave it unchanged. This is the case for our authentication-based scheme, where purified evaluation of the program for the point function for the point p on the point p will not cause any change, as the evaluation produces the correct outcome with certainty. So, in this case, it is possible to correctly evaluate on p an arbitrary number of times. It follows that to give a meaningful answer on how many times a program state can be used, we must specify how the inputs to the program are selected. We believe it is reasonable to sample them according to the same distribution for which correctness is guaranteed.

We sketch the proof of our claim. It follows from one concentration inequality, one application of the classical union bound, and one application of Gao's quantum union bound. First: A simple concentration inequality shows that if the expectation, over the choice of inputs, that the program produces the correct output is $1-\eta$, then, with probability at least $1-\sqrt{\eta}$, a randomly chosen input will be evaluated to the correct output with probability at least $1-\sqrt{\eta}$. Second: By the classical union bound, the probability that n sampled inputs are correctly evaluated with probability at least $1-\sqrt{\eta}$ is at least $1-n\sqrt{\eta}$. Third: We can model the evaluation of the program state as projective measurements where the input to the program determines the measurement. Given n measurements, each producing the correct outcome with probability $1-\sqrt{\eta}$ on the original state, Gao's quantum union bound [16] yields that the sequential application of all of these measurements will all give the correct answers with probability at least $1-4n\sqrt{\eta}$. Multiplying this with the probability that all chosen inputs satisfy the necessary condition yields our bound of $(1-n\sqrt{\eta})(1-4n\sqrt{\eta})$.

4.2 Malicious-Malicious Security and Correctness

In the full version, we show that any copy protection scheme for point functions that is secure in the malicious-malicious setting but only satisfies correctness with respect to the distribution family $\{T_p^{(1/2)}\}_p$, in which $T_p^{(1/2)}$ samples p with probability $1/2$ and all other strings uniformly, can be combined with a pairwise independent permutation family [21] to get a scheme that is still secure in the malicious-malicious setting but is also correct with respect to any distribution. We recall that the malicious-malicious security setting is the standard security definition considered in previous works, and correctness with respect to any distribution is the standard notion of correctness. Thus, our construction given in Section 5, while it has its advantages, falls short of achieving the standard security and correctness notions by being secure only in the honest-malicious setting, and by being correct only with respect to $\{T_p^{(1/2)}\}_p$. Our results (Theorem 1) show that solving the former problem would also solve the latter.

Theorem 1. *If there exists a copy protection scheme for point functions which is ϵ-malicious-malicious secure with respect to the uniform distribution on points R and challenge distribution $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$ and η-correct with respect to the distribution family $\{T_p^{(1/2)}\}_p$, then there exists a copy protection scheme for point functions which is ϵ-malicious-malicious secure with respect to the distributions R and $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$ and 2η-correct with respect to any distribution.*

4.3 Secure Software Leasing and Honest-Malicious Copy Protection

Following Figure 1, we outline how an honest-malicious copy protection scheme for some set of functions $\mathcal{C}$ can be used to create an SSL scheme for $\mathcal{C}$ (Theorem 2). Specifically, we use a copy protection scheme for a set of Boolean circuits $\mathcal{C}$ on n-bits that is correct with respect to *two* families of distributions, $\{T_C\}_{C\in\mathcal{C}}$ and $\{T'_C\}_{C\in\mathcal{C}}$, and honest-malicious secure with respect to the circuit distribution D on $\mathcal{C}$, and the challenge distributions $\{T'_C \times T''_C\}_{C\in\mathcal{C}}$, to construct an SSL scheme for $\mathcal{C}$ that is correct with respect to $\{T_C\}_{C\in\mathcal{C}}$ and secure with respect to D and $\{T''_C\}_{C\in\mathcal{C}}$. Here, $T'_C \times T''_C$ denotes the product distribution of the two distributions T'_C and T''_C, which are both distributions on $\{0,1\}^n$.

Let $\mathsf{CP} = (\mathsf{CP.Protect}, \mathsf{CP.Eval})$ be a copy protection scheme for a set of n-bit Boolean circuits $\mathcal{C}$. We define the secure software leasing scheme SSL for $\mathcal{C}$ as:

$\mathsf{SSL.Gen}$: Output an empty secret key $\mathsf{sk} = \emptyset$.
$\mathsf{SSL.Lease}(C)$: As the secret key is empty, the only input is the circuit C. On input C, output $\rho = \mathsf{CP.Protect}(C)$.
$\mathsf{SSL.Eval}(\rho, x)$: On input $\rho \in \mathcal{D}(\mathsf{Y})$ and $x \in \{0,1\}^n$, run $\mathsf{CP.Eval}$.
$\mathsf{SSL.Verify}(C, \sigma)$: As the secret key is empty, the only inputs are the circuit C and a state $\sigma \in \mathsf{Y}$. Sample $x \leftarrow T'_C$ and output 1 if and only if $\mathsf{CP.Eval}(\sigma, x)$ is $C(x)$.

Formally, we obtain the following.

Theorem 2. *Suppose the scheme CP is a copy protection scheme for circuits $\mathcal{C}$, that is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$, η-correct with respect to $\{T'_C\}_{C\in\mathcal{C}}$, and ϵ-honest-malicious secure with respect to the distribution D on $\mathcal{C}$ and challenge distributions $\{T'_C \times T''_C\}_{C\in\mathcal{C}}$. Then the scheme SSL, constructed from CP as described above, is an SSL scheme for $\mathcal{C}$ that is η-correct with respect to $\{T_C\}_{C\in\mathcal{C}}$ and ϵ-secure with respect to the distributions D and $\{T''_C\}_{C\in\mathcal{C}}$.*

The proofs for correctness and security are given in the full version. Correctness of SSL follows from the correctness of CP directly as the encoding and evaluating procedures for the programs are the same.

The main intuition for the security proof is to map the honest evaluation in the scheme CP to the Lessor's verification procedure in the scheme SSL. The ϵ-correctness of $\mathsf{CP.Eval}$ ensures that the verification is accepted with sufficiently high probability. Next, we map the malicious user Charlie's ($\mathcal{A}_2$) evaluation in $\mathsf{PiratingGame}$ to the adversary's evaluation in $\mathsf{SSLGame}$. Assuming that CP is secure, we can bound Charlie's probability of guessing the right answer, which

in turn bounds the adversary's probability of guessing the right answer. Putting it together, we can conclude that the corresponding SSL scheme SSL is secure.

We remark that this previous proof does not make any assumptions about the abilities of the adversaries. Hence, if the copy protection scheme CP achieves statistical security guarantees, then so does the corresponding SSL scheme SSL.

4.4 Secure Software Leasing of Compute-and-Compare Circuits

In this section we present a restatement of a theorem due to [11], which states that an SSL scheme for point functions that is ϵ-secure with respect to a family of distributions can be modified to get an SSL scheme for compute-and-compare programs that is also ϵ-secure with respect to a related family of distributions. We state this result with a more precise relationship between the distributions used for the point functions and the compute-and-compare programs.

Let F denote any set of functions from $\{0,1\}^n$ to $\{0,1\}^m$. We then let the set $\mathcal{F} = \{(f,y) : f \in F, y \in \{0,1\}^m\}$ be the set of compute-and-compare circuits for F, where as with point functions, we conflate (f,y) with a circuit CC^f_y for the function that outputs 1 on input x if and only if $f(x) = y$.

Let $\mathsf{PF} = (\mathsf{PF.Gen}, \mathsf{PF.Lease}, \mathsf{PF.Eval}, \mathsf{PF.Verify})$ be an SSL scheme for m-bit point functions. Using the same construction as in [11], we obtain an SSL scheme for compute-and-compare functions $\mathcal{F}$, CC, from the scheme PF.

Formally, we show the following theorem.

Theorem 3. *We fix the following distributions.*

- *D: A distribution over compute-and-compare functions CC^f_y, or equivalently, over $(f,y) \in \mathcal{F}$. Fixing a function $f \in F$ induces a marginal distribution D_f over $y \in \{0,1\}^m$, or equivalently, over m-bit point functions P_y.*
- *$\{T^{CC}_{f,y}\}_{f,y}$ and $\{D^{CC}_{f,y}\}_{f,y}$: Families of distributions over inputs $x \in \{0,1\}^n$ to compute-and-compare functions CC^f_y.*
- *$\{T^{PF}_{f,y}\}_{f,y}$ and $\{D^{PF}_{f,y}\}_{f,y}$: Families of distributions over inputs $z \in \{0,1\}^m$ to m-bit point functions P_y, where $T^{PF}_{f,y}$ is defined from $T^{CC}_{f,y}$ by sampling $x \leftarrow T^{CC}_{f,y}$ and outputting $f(x)$; and $D^{PF}_{f,y}$ is defined similarly from $D^{CC}_{f,y}$.*

Suppose that PF is a secure software leasing scheme for point functions such that, for every $f \in F$, PF is η-correct with respect to the distribution family $\{T^{PF}_{f,y}\}_{y\in\{0,1\}^m}$ and ϵ_f-secure with respect to the circuit distribution D_f and challenge distributions $\{D^{PF}_{f,y}\}_{y\in\{0,1\}^m}$ where

$$\epsilon_f = \left(p^{triv}_{D,\{D^{CC}_{f,y}\}_{(f,y)}} - p^{triv}_{D_{f^*},\{D^{PF}_{f^*,y}\}_y}\right) + \epsilon. \tag{16}$$

Then the scheme CC, constructed from PF as described above, is an SSL scheme for compute-and-compare programs in $\mathcal{F}$ that is η-correct with respect to the family $\{T^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$ and ϵ-secure with respect to program distribution D and challenge distributions $\{D^{CC}_{f,y}\}_{(f,y)\in\mathcal{F}}$.

The proof of correctness follows directly from definitions and the security proof follows the same lines as the one presented in [11]. For completeness, the construction and proofs are given in the full version.

5 Authentication-based Copy Protection Scheme

In this section, we show how to construct a copy protection scheme for point functions, with honest-malicious security, from a total authentication scheme.

Recall that we assume that our circuits are searchable, which, for point functions, implies that there is an efficient algorithm which can produce the point p from a circuit which computes its point function. Thus, we will freely identify circuits for the point function P_p simply with p. Specifically, our copy protection scheme will take as input a point p instead of a circuit.

5.1 Construction and Correctness

Let $\mathsf{QAS} = (\mathsf{QAS.Auth}, \mathsf{QAS.Ver})$ be an ϵ-total quantum authentication scheme, as in Definition 1, with $\epsilon \leq \frac{1}{2}$ for a message space M of dimension greater than or equal to two with key set $\mathcal{K} = \{0,1\}^n$. Fix some state $|\psi\rangle \in \mathsf{M}$.

We recall that we assume that for every key k, the action of $\mathsf{QAS.Auth}$ with this key can be modeled by an isometry $A_k : \mathsf{M} \to \mathsf{Y}$. Note that since A_k is an isometry, $A_k A_k^\dagger$ is the projector onto $\mathrm{im}(A_k)$. Further, let $V_k : \mathsf{Y} \to \mathsf{MFX}$ be an isometry which purifies the CPTP map $\mathsf{QAS.Ver}_k$ defined in Eq. (6), where the register X corresponds to the Hilbert space used for this purification. To simplify our notation, we will absorb X into the flag register, which we no longer assume to be two-dimensional. We can still assume that there is a unique accepting state $|\mathrm{Acc}\rangle \in \mathsf{F}$.[15] Thus, from here on, we assume that $V_k : \mathsf{Y} \to \mathsf{MF}$ is an isometry, and F has dimension at least two (but possibly larger) with $|\mathrm{Acc}\rangle$ the accepting state, and all orthogonal states rejecting.

Finally, we will write $\overline{V}_k = (\langle \mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}})V_k$ to denote the map which applies the verification, but only outputs the state corresponding to the verification procedure accepting, corresponding to the procedure $\mathsf{QAS.Ver}'_k$ described in Section 2.2. Then note that $\overline{V}_k = A_k^\dagger$.

From this authentication scheme and fixed state $|\psi\rangle$, which can be assumed without loss of generality to be $|0\rangle$, we construct a copy protection scheme for point functions of length n, AuthCP, as follows:

$\mathsf{AuthCP.Protect}(p)$: On input $p \in \{0,1\}^n$, do the following:

1. Output $A_p |\psi\rangle$.

$\mathsf{AuthCP.Eval}(\sigma, x)$: On input $\sigma \in \mathcal{D}(\mathsf{Y})$ and $x \in \{0,1\}^n$, do the following:

1. Compute $\xi = V_x \sigma V_x^\dagger$. Recall that ξ is a state on registers F, the flag register, and M, the message register.
2. Measure the F register of ξ in $\{|\mathrm{Acc}\rangle\langle\mathrm{Acc}|, I - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|\}$. If the outcome obtained is "Acc", output 1. Otherwise, output 0.

[15] This follows from correctness, since for every state $|\psi\rangle$, we necessarily have $V_k A_k |\psi\rangle = |\mathrm{Acc}\rangle_{\mathsf{F}} |\psi\rangle_{\mathsf{M}} |X_\psi\rangle_{\mathsf{X}}$ for some state $|X_\psi\rangle$, and by the fact that $V_k A_k$ must preserve inner products, we necessarily have $|X_\psi\rangle = |X\rangle$ independent of $|\psi\rangle$. Thus, we can let $|\mathrm{Acc}\rangle_{\mathsf{F}} |X\rangle_{\mathsf{X}}$ be the accepting state on FX.

We recall that correctness is parametrized by a family of input distributions to each point function, and security is parametrized by a distribution on the possible functions to be encoded and by a family of distributions on challenges to send the users Bob and Charlie. Our correctness and security are proven with respect to the following distributions:

- Our correctness will be with respect to the distribution $T_p^{(1/2)}$, as defined in Definition 12, which we recall is the distribution on $\{0,1\}^n$ in which p is sampled with probability $1/2$, and all other strings are sampled with probability $\frac{1}{2(2^n-1)}$.
- In our security proof, we will assume that the point p of the challenge function is chosen uniformly at random. This corresponds to the distribution R given in Definition 11.
- If the challenge function is specified by the point p, the challenges will be sampled independently according to the distribution $T_p^{(1/2)}$. We will refer to this as $T_p^{(1/2)} \times T_p^{(1/2)}$.

We first prove the correctness of the scheme AuthCP.

Theorem 4. *If the scheme* QAS *is an ϵ-total authentication scheme, then the scheme* AuthCP *described above is ϵ-correct with respect to the family of distributions $\{T_p^{(1/2)}\}_p$.*

Proof. For all $p \in \{0,1\}^n$, it suffices to compute a lower bound on

$$\frac{1}{2}\left\|\overline{V}_p A_p \left|\psi\right\rangle\right\|^2 + \frac{1}{2}\cdot\frac{1}{2^n-1}\sum_{\substack{x\in\{0,1\}^n \\ x\neq p}}\left(1-\left\|\overline{V}_x A_p \left|\psi\right\rangle\right\|^2\right). \tag{17}$$

By the correctness of the authentication scheme, we have that $\left\|\overline{V}_p A_p \left|\psi\right\rangle\right\|^2 = 1$. On the other hand, by Lemma 2, we have that

$$\sum_{\substack{x\in\{0,1\}^n \\ x\neq p}}\left\|\overline{V}_x A_p \left|\psi\right\rangle\right\|^2 \leq 2^n\cdot 2\epsilon - 1 \tag{18}$$

by expanding the expectation and removing the term corresponding to $x = p$. Thus, a lower bound for Eq. (17) is given by

$$\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{2^n-1}\left(2^n-1-\sum_{\substack{x\in\{0,1\}^n \\ x\neq p}}\left\|\overline{V}_x A_p \left|\psi\right\rangle\right\|^2\right) \geq \frac{1}{2}+\frac{1}{2}\left(1-\frac{2^n\cdot 2\epsilon-1}{2^n-1}\right) \geq 1-\epsilon,$$

as long as $\epsilon \leq 1/2$, and so the scheme is ϵ-correct with respect to the given distribution family. □

5.2 Honest-Malicious Security

In this section, we prove the security of the scheme AuthCP in the honest-malicious setting. Formally, we prove the following theorem.

Theorem 5. *If the scheme* QAS *is an ϵ-total authentication scheme, then the scheme* AuthCP *described above is $(\frac{3}{2}\epsilon + \sqrt{2\epsilon})$-honest-malicious secure with respect to the uniform distribution R on point functions and challenge distributions $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_{p\in\{0,1\}^n}$, where R and $T_p^{(1/2)}$ are as defined in Definition 11 and Definition 12.*

In fact, we can prove security with respect to a slightly more general set of challenge distributions. If we let $T_p^{(r)}$ be the distribution that samples p with probability r, and any other point uniformly, then for any $r \in [1/2, 1]$, our proof holds when Bob's input is chosen according to $T_p^{(r)}$ and Charlie's input is chosen according to $T_p^{(1/2)}$. (See Remark 1 following the proof of Theorem 5). If Bob gets the point with probability less than $1/2$, then it becomes easier for the adversary to win. Pete can simply send the program to Charlie, and give Bob a maximally mixed state. In that case, Bob will probably output 0, which is correct more than $1/2$ the time.

For the challenge distributions R and $\{T_p^{(1/2)} \times T_p^{(1/2)}\}_p$, it is easy to see that Charlie's maximum guessing probability if he has no interaction with Pete, against which we measure security (see Definition 6), is $p^{\text{marg}} = 1/2$. We will use this fact in our security proof, which could likely be generalized to other distributions of Charlie's challenge with a different value of p^{marg}, but we do not analyze such cases.

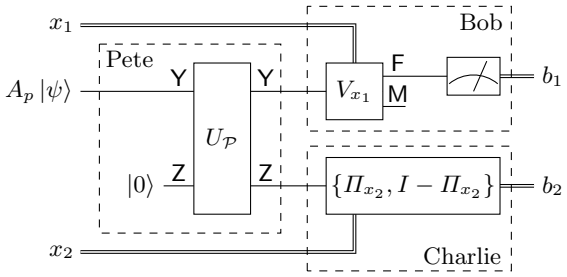

Fig. 4. The pirating game specified to the AuthCP scheme.

The idea of the proof is the following. In the setting of the scheme AuthCP, the pirating game $\mathsf{PiratingGame}_{\hat{\mathcal{A}},\mathsf{AuthCP}}$ (see Fig. 2) that an honest-malicious adversary must win is expressed in Fig. 4. Without loss of generality, we can assume Pete's behaviour is modeled by a unitary $U_{\mathcal{P}}$ on the space YZ for an arbitrary auxiliary space Z initialized to a fixed state, which we will denote $|0\rangle$. (Note that this state can be composed of more than one qubit.)

Since the adversary is honest-malicious, we can assume that Bob is honestly evaluating the program, meaning he runs the verification procedure of the underlying authentication scheme, using the point he receives as the key, on the register Y, outputting 1 if and only if the flag register F is measured as "Acc".

Charlie's behaviour can be arbitrary, but without loss of generality, we can assume that it is specified by a family of two-outcome measurements on Z, $\{\Pi_x, I-\Pi_x\}_{x\in\{0,1\}^n}$. Charlie uses his challenge input x_2 to select a measurement to perform to obtain his output b_2.

We will break the proof into two cases. First, consider the case where $x_1 = p$. We can consider Pete's output in two orthogonal parts:

$$U_{\mathcal{P}}(A_p \left|\psi\right\rangle \otimes \left|0\right\rangle) = \left|\Gamma^p_{\mathrm{Acc}}\right\rangle + \left|\Gamma^p_{\mathrm{Rej}}\right\rangle, \tag{19}$$

where $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ is the part of the state that leads to Bob outputting 1 on input p, which is the correct bit for Bob to produce in this case. That is, $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ is the projection of Pete's output onto states where the Y register is supported on the image of A_p. When $x_1 = p$, only $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ contributes to a winning outcome. We show (Lemma 5) that this state is close (on average over p) to a state of the form $A_p \left|\psi\right\rangle\!\left\langle\psi\right| A_p^\dagger \otimes \xi_{\mathsf{Z}}$ for some subnormalized state ξ independent of p. Since Charlie's input is essentially independent of p, his winning probability is not much more than $1/2$, so the total winning probability in this case is not much more than $1/2$ (Lemma 4), which is scaled down by the trace of the subnormalized state ξ, representing the fact that the probability that Bob outputs the correct bit is $\left\|\left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\|^2$.

The other case is when $x_1 \neq p$. In that case, we need to consider the contribution of both terms $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ and $\left|\Gamma^p_{\mathrm{Rej}}\right\rangle\!\!\right\rangle$, as well as their cross term. We can bound the contribution of the first term to just over $\frac{1}{2}\operatorname{Tr}(\xi)$ because Charlie's input is close to p-independent. As for the contribution of the second term, in the worst case, the second term is of the form $\alpha \left|0\right\rangle_{\mathsf{Y}} \otimes A_p \left|\psi\right\rangle$ for some scaling factor α. This corresponds to the strategy that Pete just sends Charlie the program. Charlie can evaluate the program and be correct with probability close to 1, and Bob will output 0 with probability close to 1, which is the correct bit in this case, since $x_1 \neq p$. So we trivially upper bound the contribution of this term by $\left\|\left|\Gamma^p_{\mathrm{Rej}}\right\rangle\!\!\right\rangle\right\|^2$. However, as this increases, the size of $\left|\Gamma^p_{\mathrm{Acc}}\right\rangle$ and thus $\operatorname{Tr}(\xi)$ decreases, so the probability of being correct in the $x_1 = p$ case goes down. We find that the total contribution, ignoring the cross term, is at most negligibly more than $1/2$. Finally, we show that the cross-term is negligible by the correctness of the scheme AuthCP.

We first state and prove the necessary lemmas, before formalizing the above argument. The following lemma is simply stating that if Charlie gets an input state that is independent of the point p, then his guess as to whether $x_2 = p$ will be independent of p, and so will be correct with probability $1/2$.

Lemma 4. *Suppose p is chosen uniformly at random, and $x_2 \leftarrow T_p^{(1/2)}$, so that with probability $1/2$, $x_2 = p$, and otherwise x_2 is uniform on $\{0,1\}^n \setminus \{p\}$. Let*

$\Pi^1_{x_2} = \Pi_{x_2}$ *and* $\Pi^0_{x_2} = I - \Pi_{x_2}$*, so* $\Pi^{P_p(x_2)}_{x_2} = \Pi_{x_2}$ *when* $x_2 = p$*, and otherwise* $\Pi^{P_p(x_2)}_{x_2} = I - \Pi_{x_2}$*. Then, for any density matrix* σ*,* $\mathbb{E}_{p,x_2} \operatorname{Tr}\left(\Pi^{P_p(x_2)}_{x_2}\sigma\right) = \frac{1}{2}$*.*

Proof. It suffices to compute:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p,x_2} \operatorname{Tr}\left(\Pi^{P_p(x_2)}_{x_2}\sigma\right) \\
&= \frac{1}{2^n} \sum_{p\in\{0,1\}^n} \left(\frac{1}{2}\operatorname{Tr}(\Pi_p\sigma) + \frac{1}{2}\frac{1}{2^n-1} \sum_{x_2\neq p} \operatorname{Tr}((I-\Pi_{x_2})\sigma) \right) \\
&= \frac{1}{2}\frac{1}{2^n} \sum_{p\in\{0,1\}^n} \operatorname{Tr}(\Pi_p\sigma) + \frac{1}{2}\left(1 - \frac{1}{2^n}\sum_{x_2\in\{0,1\}^n} \operatorname{Tr}(\Pi_{x_2}\sigma)\right) = \frac{1}{2}. \qquad \square
\end{aligned}$$

The following lemma tells us that in the part of Pete's output that will be accepted by Bob in the $x_1 = p$ case, Bob's input from Pete is essentially $A_p|\psi\rangle$, and Charlie's input from Pete is almost independent of p. Recall that A_p is an isometry, so $A_pA_p^\dagger$ is the projector onto $\operatorname{im}(A_p)$.

Lemma 5. *Let* $|\Gamma^p_{Acc}\rangle_{\mathsf{YZ}} = (A_pA_p^\dagger \otimes I_{\mathsf{Z}})U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle)$ *be the projection of Pete's output onto states supported on* $im(A_p)$ *in the* Y *register. Then, there exists a subnormalized state* $\xi \in \mathcal{D}(\mathsf{Z})$ *such that*

$$\mathop{\mathbb{E}}_{p} \Delta\left(|\Gamma^p_{Acc}\rangle\langle\Gamma^p_{Acc}|, A_p|\psi\rangle\langle\psi|A_p^\dagger \otimes \xi\right) \leq \epsilon. \tag{20}$$

Proof. By the security of the total authentication scheme, there exists a completely positive trace non-increasing map $\Psi : \mathcal{L}(\mathsf{Z}) \to \mathcal{L}(\mathsf{Z})$ such that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{p} |p\rangle\langle p| \otimes (\overline{V}_p \otimes I_{\mathsf{Z}})(U_{\mathcal{P}})_{\mathsf{YZ}}(A_p|\psi\rangle\langle\psi|A_p^\dagger \otimes |0\rangle\langle 0|_{\mathsf{Z}})(U_{\mathcal{P}})^\dagger_{\mathsf{YZ}}(\overline{V}_p^\dagger \otimes I_{\mathsf{Z}}) \\
\approx_\epsilon &\mathop{\mathbb{E}}_{p} |p\rangle\langle p| \otimes (\overline{V}_pA_p)|\psi\rangle\langle\psi|_{\mathsf{M}} (A_p^\dagger\overline{V}_p^\dagger) \otimes \Psi(|0\rangle\langle 0|).
\end{aligned} \tag{21}$$

Using the fact that $\overline{V}_p = A_p^\dagger = A_p^\dagger(A_pA_p^\dagger)$ (that is, project onto states in the image of A_p, and then invert A_p), we have

$$\begin{aligned}
(\overline{V}_p \otimes I_{\mathsf{Z}})U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle_{\mathsf{Z}}) &= (\overline{V}_p \otimes I_{\mathsf{Z}})(A_pA_p^\dagger \otimes I_{\mathsf{Z}})U_{\mathcal{P}}(A_p|\psi\rangle \otimes |0\rangle_{\mathsf{Z}}) \\
&= (\overline{V}_p \otimes I_{\mathsf{Z}})|\Gamma^p_{\mathrm{Acc}}\rangle.
\end{aligned}$$

Then by Lemma 1, and letting $\xi = \Psi(|0\rangle\langle 0|)$, we can continue from Eq. (21) to get:

$$\begin{aligned}
\mathop{\mathbb{E}}_{p} \Delta\left(\overline{V}_p|\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}|\overline{V}_p^\dagger, \overline{V}_pA_p|\psi\rangle\langle\psi|A_p^\dagger\overline{V}_p^\dagger \otimes \xi\right) &\leq \epsilon \\
\mathop{\mathbb{E}}_{p} \Delta\left(|\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}|, A_p|\psi\rangle\langle\psi|A_p^\dagger \otimes \xi\right) &\leq \epsilon,
\end{aligned}$$

where we used the fact that $|\Gamma^p_{\mathrm{Acc}}\rangle$ and $A_p|\psi\rangle$ are both orthogonal to the kernel of $\overline{V}_p$, so the isometry $\overline{V}_p$ preserves the distance between them. $\square$

We now proceed to prove our main theorem of this section, Theorem 5.

Proof of Theorem 5. For a fixed p, x_1 and x_2, let q_1^{p,x_2} be the adversary's winning probability when $x_1 = p$, and let q_0^{p,x_1,x_2} be the winning probability when $x_1 \neq p$. Then the total winning probability is given by

$$\frac{1}{2} \mathop{\mathbb{E}}_{\substack{p\leftarrow R,\\ x_2\leftarrow T_p^{(1/2)}}} q_1^{p,x_2} + \frac{1}{2} \mathop{\mathbb{E}}_{\substack{p\leftarrow R,\\ x_1\leftarrow\{0,1\}^n\setminus p,\\ x_2\leftarrow T_p^{(1/2)}}} q_0^{p,x_1,x_2}. \tag{22}$$

If $|\Gamma^p\rangle := U_{\mathcal{P}}(A_p |\psi\rangle \otimes |0\rangle)$ is Pete's output for a fixed p, and $\Pi_{x_2}^{P_p(x_2)}$ is defined to be Π_p when $x_2 = p$ and $I - \Pi_{x_2}$ otherwise, we have that

$$\begin{aligned} q_1^{p,x_2} &= \left\| (|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}} \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})(V_p \otimes \mathbb{1}_{\mathsf{Z}}) |\Gamma^p\rangle \right\|^2 \\ \text{and} \quad q_0^{p,x_1,x_2} &= \left\| ((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes I_{\mathsf{M}} \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})(V_{x_1} \otimes \mathbb{1}_{\mathsf{Z}}) |\Gamma^p\rangle \right\|^2. \end{aligned}$$

We will upper bound q_1^{p,x_2} and q_0^{p,x_1,x_2} separately.

Recall that we can write Pete's output as

$$|\Gamma^p\rangle = |\Gamma^p_{\mathrm{Acc}}\rangle + \left|\Gamma^p_{\mathrm{Rej}}\right\rangle, \tag{23}$$

where

$$\begin{aligned} |\Gamma^p_{\mathrm{Acc}}\rangle &= (A_p A_p^\dagger \otimes I_{\mathsf{Z}}) |\Gamma^p\rangle \\ \text{and} \quad \left|\Gamma^p_{\mathrm{Rej}}\right\rangle &= ((I_{\mathsf{Y}} - A_p A_p^\dagger) \otimes I_{\mathsf{Z}}) |\Gamma^p\rangle. \end{aligned} \tag{24}$$

The $x_1 = p$ case. We begin by upper bounding q_1^{p,x_2}. We first show there is no contribution from the second term:

$$\begin{aligned} &(|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}} \otimes \Pi_{x_2}^{P_p(x_2)})(V_p \otimes I_{\mathsf{Z}}) \left|\Gamma^p_{\mathrm{Rej}}\right\rangle \\ &= (|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}} \otimes I_{\mathsf{M}} \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})(V_p(I_{\mathsf{Y}} - A_p A_p^\dagger) \otimes I_{\mathsf{Z}}) |\Gamma^p\rangle \\ &= 0 \end{aligned} \tag{25}$$

because

$$\begin{aligned} (\langle\mathrm{Acc}| \otimes I_{\mathsf{M}}) V_p (I_{\mathsf{Y}} - A_p A_p^\dagger) &= \overline{V}_p (I_{\mathsf{Y}} - A_p A_p^\dagger) \\ &= A_p^\dagger A_p A_p^\dagger (I_{\mathsf{Y}} - A_p A_p^\dagger) \\ &= 0. \end{aligned} \tag{26}$$

Above we used the fact that $\overline{V}_p = A_p^\dagger = A_p^\dagger (A_p A_p^\dagger)$ which is to say that $\overline{V}_p$ simply projects onto states in the image of A_p, and then inverts A_p. Thus (omitting

implicit tensored identities):

$$\begin{aligned} q_1^{p,x_2} &= \mathrm{Tr}\left((|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_F \otimes \Pi_{x_2}^{P_p(x_2)})V_p\,|\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}|\,V_p^\dagger\right) \\ &= \mathrm{Tr}\left(V_p^\dagger(|\mathrm{Acc}\rangle\langle\mathrm{Acc}|_F \otimes \Pi_{x_2}^{P_p(x_2)})V_p(A_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \xi + \delta_p)\right) \\ &\le \mathrm{Tr}\left(V_p^\dagger\,|\mathrm{Acc}\rangle\langle\mathrm{Acc}|\,V_pA_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \Pi_{x_2}^{P_p(x_2)}\xi\right) \\ &\quad + \Delta\left(|\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}|\,, A_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \xi\right) \\ &= \mathrm{Tr}\Big(\Pi_{x_2}^{P_p(x_2)}\xi\Big) + \Delta\left(|\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}|\,, A_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \xi\right), \end{aligned} \tag{27}$$

where $\delta_p = |\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}| - A_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \xi$.

By Lemma 4, we have $\mathrm{Tr}(\xi)\,\mathbb{E}_{p,x_2}\,\mathrm{Tr}\left(\Pi_{x_2}^{P_p(x_2)}\frac{\xi}{\mathrm{Tr}(\xi)}\right) = \mathrm{Tr}(\xi)/2$. Combining this with Lemma 5, we conclude with:

$$\mathop{\mathbb{E}}_{p,x_2} q_1^{p,x_2} \le \frac{\mathrm{Tr}(\xi)}{2} + \epsilon. \tag{28}$$

The $x_1 \neq p$ case: We will analyze the probability in three parts, as follows:

$$\begin{aligned} q_0^{p,x_1,x_2} &= \left\|((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes (\Pi_{x_2}^{P_p(x_2)})_{\mathsf{Z}})V_{x_1}\,|\Gamma^p\rangle\right\|^2 \\ &\le \underbrace{\left\|((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes \Pi_{x_2}^{P_p(x_2)})V_{x_1}\,|\Gamma^p_{\mathrm{Acc}}\rangle\right\|^2}_{=T_1^{p,x_1,x_2}} \\ &+ \underbrace{\left\|((I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}}) \otimes \Pi_{x_2}^{P_p(x_2)})V_{x_1}\left|\Gamma^p_{\mathrm{Rej}}\right\rangle\right\|^2}_{=T_2^{p,x_1,x_2}} \\ &+ \underbrace{2\left|\left\langle\Gamma^p_{\mathrm{Rej}}\right|(V_{x_1}^\dagger(I_{\mathsf{F}} - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|_{\mathsf{F}})V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)})\,|\Gamma^p_{\mathrm{Acc}}\rangle\right|}_{=T_{\mathrm{cross}}^{p,x_1,x_2}}. \end{aligned} \tag{29}$$

We begin with the first term, whose analysis is similar to the $x_1 = p$ case. We have:

$$\begin{aligned} T_1^{p,x_1,x_2} &= \mathrm{Tr}\left((V_{x_1}^\dagger(I - |\mathrm{Acc}\rangle\langle\mathrm{Acc}|)V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)})(A_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \xi + \delta_p)\right) \\ &\le \mathrm{Tr}\Big(\Pi_{x_2}^{P_p(x_2)}\xi\Big) + \Delta(|\Gamma^p_{\mathrm{Acc}}\rangle\langle\Gamma^p_{\mathrm{Acc}}|\,, A_p\,|0\rangle\langle 0|\,A_p^\dagger \otimes \xi). \end{aligned} \tag{30}$$

Thus, just as we concluded with Eq. (28), we can conclude

$$\mathop{\mathbb{E}}_{p,x_1,x_2} T_1^{p,x_1,x_2} \le \frac{\mathrm{Tr}(\xi)}{2} + \epsilon, \tag{31}$$

again, by Lemma 4 and Lemma 5.

For the second term, we will use the naive bound:

$$\begin{aligned} T_2^{p,x_1,x_2} &\leq \left\| \left| \Gamma^p_{\mathrm{Rej}} \right\rangle \right\|^2 \\ &= 1 - \| |\Gamma^p_{\mathrm{Acc}}\rangle \|^2 \\ &\leq 1 - \mathrm{Tr}\big(A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi\big) + \Delta\left(|\Gamma^p_{\mathrm{Acc}}\rangle\!\langle\Gamma^p_{\mathrm{Acc}}|, A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi\right) \\ &= 1 - \mathrm{Tr}(\xi) + \Delta\left(|\Gamma^p_{\mathrm{Acc}}\rangle\!\langle\Gamma^p_{\mathrm{Acc}}|, A_p \left|0\right\rangle\!\left\langle 0\right| A_p^\dagger \otimes \xi\right). \end{aligned} \tag{32}$$

Then by Lemma 5, we have

$$\mathop{\mathbb{E}}_{p,x_1,x_2} T_2^{p,x_1,x_2} \leq 1 - \mathrm{Tr}(\xi) + \epsilon. \tag{33}$$

Finally, we upper bound the cross-term. The idea is that $|\Gamma^p_{\mathrm{Acc}}\rangle$ and $\left|\Gamma^p_{\mathrm{Rej}}\right\rangle$ are orthogonal in the Y register. This is, of course, also true once we apply $\Pi_{x_2}^{P_p(x_2)}$ to the Z register. Applying the projector $V_{x_1}^\dagger(I_{\mathsf{F}} - |\mathrm{Acc}\rangle\!\langle\mathrm{Acc}|_{\mathsf{F}})V_{x_1}$ to the Y register could change this, however, we will argue that, by correctness of the scheme, this projector cannot change the state $|\Gamma^p_{\mathrm{Acc}}\rangle$ very much, because its first register is in $\mathrm{im}(A_p)$, and trying to decode with a different key, $x_1 \neq p$, should result in rejection with high probability. More formally, we have:

$$\begin{aligned} &T^{p,x_1,x_2}_{\mathrm{cross}} \\ &= 2\left|\left\langle \Gamma^p_{\mathrm{Rej}}\right| (I_{\mathsf{Y}} \otimes \Pi_{x_2}^{P_p(x_2)}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle - \left\langle \Gamma^p_{\mathrm{Rej}}\right| (V_{x_1}^\dagger \left|\mathrm{Acc}\right\rangle\!\left\langle\mathrm{Acc}\right| V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right| \\ &= 2\left|\left\langle \Gamma^p_{\mathrm{Rej}}\right| (V_{x_1}^\dagger \left|\mathrm{Acc}\right\rangle\!\left\langle\mathrm{Acc}\right| V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right| \\ &\leq 2\left\|(\langle\mathrm{Acc}| V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)}) \left|\Gamma^p_{\mathrm{Rej}}\right\rangle\right\| \cdot \left\|(\langle\mathrm{Acc}| V_{x_1} \otimes \Pi_{x_2}^{P_p(x_2)}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\| \\ &\leq 2\|(\langle\mathrm{Acc}| V_{x_1} \otimes I_{\mathsf{Z}}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\| = 2\left\|(\overline{V}_{x_1} \otimes I_{\mathsf{Z}}) \left|\Gamma^p_{\mathrm{Acc}}\right\rangle\right\|, \end{aligned}$$

where we use the orthogonality on the Y register of $|\Gamma^p_{\mathrm{Acc}}\rangle$ and $\left|\Gamma^p_{\mathrm{Rej}}\right\rangle$ to obtain the second equality and the Cauchy-Schwarz inequality to obtain the first inequality. Since $|\Gamma^p_{\mathrm{Acc}}\rangle$ is supported on $\mathrm{im}(A_p)$ in the first register, it has a Schmidt decomposition of the form:

$$|\Gamma^p_{\mathrm{Acc}}\rangle = \sum_\ell \beta_\ell (A_p \left|u_\ell\right\rangle)_{\mathsf{Y}} \otimes \left|v_\ell\right\rangle_{\mathsf{Z}}. \tag{34}$$

Taking the expectation and applying Jensen's inequality, we have:

$$\begin{aligned} \mathop{\mathbb{E}}_{p,x_1,x_2} T^{p,x_1,x_2}_{\mathrm{cross}} &\leq 2 \mathop{\mathbb{E}}_{p,x_1} \sqrt{\sum_\ell |\beta_\ell|^2 \left\|\overline{V}_{x_1} A_p \left|u_\ell\right\rangle\right\|^2} \\ &\leq 2\sqrt{\sum_\ell |\beta_\ell|^2 \mathop{\mathbb{E}}_{p,x_1} \left\|\overline{V}_{x_1} A_p \left|u_\ell\right\rangle\right\|^2} \end{aligned} \tag{35}$$

We next want to appeal to Lemma 2, which implies that for any pure state $|u\rangle$ we have that $\mathbb{E}_{p,x_1\leftarrow\{0,1\}^n} \left\|\overline{V}_{x_1} A_p \left|u\right\rangle\right\|^2 \leq 2\epsilon$, however, notice that p and x_1 are

not uniformly distributed, because while p is uniform, x_1 is uniform over the set $\{0,1\}^n \setminus \{p\}$. However, since for any p we have $\left\|\overline{V}_p A_p \left|u\right\rangle\right\|^2 = 1$, we have:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{p\leftarrow\{0,1\}^n,\\ x_1\leftarrow\{0,1\}^n\setminus\{p\}}} \left\|\overline{V}_{x_1} A_p \left|u_\ell\right\rangle\right\|^2 \\
&= \frac{2^{2n}}{2^n(2^n-1)} \left(\mathop{\mathbb{E}}_{\substack{p\leftarrow\{0,1\}^n,\\ x_1\leftarrow\{0,1\}^n}} \left\|\overline{V}_{x_1} A_p \left|u_\ell\right\rangle\right\|^2 - \frac{1}{2^{2n}} \sum_{p\in\{0,1\}^n} \left\|\overline{V}_p A_p \left|u_\ell\right\rangle\right\|^2 \right) \qquad (36)\\
&\leq 2\epsilon + \frac{1}{2^n-1} 2\epsilon - \frac{1}{2^n-1}
\end{aligned}$$

which is at most 2ϵ as long as $\epsilon \leq 1/2$. Thus we can continue:

$$\begin{aligned}
\mathop{\mathbb{E}}_{p,x_1,x_2} T_{\text{cross}}^{p,x_1,x_2} &\leq 2\sqrt{\sum_\ell \left|\beta_\ell\right|^2}\sqrt{2\epsilon} \\
&= 2\sqrt{2\epsilon}.
\end{aligned} \qquad (37)$$

Combining Eq. (31), Eq. (33), and Eq. (35) into Eq. (29), we conclude the $x_1 \neq p$ case with:

$$\begin{aligned}
\mathop{\mathbb{E}}_{p,x_1,x_2} q_0^{p,x_1,x_2} &\leq \mathop{\mathbb{E}}_{p,x_1,x_2} T_1^{p,x_1,x_2} + \mathop{\mathbb{E}}_{p,x_1,x_2} T_2^{p,x_1,x_2} + \mathop{\mathbb{E}}_{p,x_1,x_2} T_{\text{cross}}^{p,x_1,x_2} \\
&\leq \frac{1}{2}\operatorname{Tr}(\xi) + \epsilon + 1 - \operatorname{Tr}(\xi) + \epsilon + 2\sqrt{2\epsilon}.
\end{aligned} \qquad (38)$$

Conclusion. We can now combine Eq. (28) and Eq. (38) to get an upper bound on the total winning probability of:

$$\begin{aligned}
&\frac{1}{2}\mathop{\mathbb{E}}_{p,x_2} q_1^{p,x_2} + \frac{1}{2}\mathop{\mathbb{E}}_{p,x_1,x_2} q_0^{p,x_1,x_2} \\
&\leq \frac{1}{2}\left(\frac{1}{2}\operatorname{Tr}(\xi) + \epsilon\right) + \frac{1}{2}\left(1 - \frac{1}{2}\operatorname{Tr}(\xi) + 2\epsilon + 2\sqrt{2\epsilon}\right) \qquad (39)\\
&= \frac{1}{2} + \frac{3}{2}\epsilon + \sqrt{2\epsilon}.
\end{aligned}$$

Noting that $p^{\text{marg}}_{R,\{T_p^{(1/2)}\otimes T_p^{(1/2)}\}_p} = \frac{1}{2}$ completes the proof. □

Remark 1. We note that if our challenge distribution instead chooses Bob's input so that $x_1 = p$ with probability r, for $r \geq 1/2$, and all other points uniformly,

then Eq. (39) would instead give us:

$$\begin{aligned}
&r \mathop{\mathbb{E}}_{p,x_2} q_1^{p,x_2} + (1-r) \mathop{\mathbb{E}}_{p,x_1,x_2} q_0^{p,x_1,x_2} \\
&\leq r\left(\frac{1}{2}\operatorname{Tr}(\xi)+\epsilon\right) + (1-r)\left(1-\frac{1}{2}\operatorname{Tr}(\xi)+2\epsilon+2\sqrt{2\epsilon}\right) \\
&= \frac{1}{2}(2r-1)\operatorname{Tr}(\xi) + 1 - r + (2-r)\epsilon + 2(1-r)\sqrt{2\epsilon} \qquad (40)\\
&\leq \frac{1}{2}(2r-1) + 1 - r + (2-r)\epsilon + 2(1-r)\sqrt{2\epsilon} \\
&= \frac{1}{2} + (2-r)\epsilon + 2(1-r)\sqrt{2\epsilon}.
\end{aligned}$$

We therefore have $((2-r)\epsilon + 2(1-r)\sqrt{2\epsilon})$-honest-malicious security under this more general challenge distribution, where Bob's input is distributed as $T_p^{(r)}$ and Charlie's input is distributed as $T_p^{(1/2)}$.

Acknowledgements. We would like to thank Christian Majenz and Martti Karvonen for related discussions. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0083, Canada's NFRF and NSERC, an Ontario ERA, and the University of Ottawa's Research Chairs program. SJ is a CIFAR Fellow in the Quantum Information Science program.

References

1. Aaronson, S.: Quantum copy-protection and quantum money. In: 24th Annual Conference on Computational Complexity—CCC 2009. pp. 229–242 (2009). https://doi.org/10.1109/CCC.2009.42
2. Aaronson, S., Christiano, P.: Quantum money from hidden subspaces. In: 44th Annual ACM Symposium on Theory of Computing—STOC 2012. pp. 41–60 (2012). https://doi.org/10.1145/2213977.2213983
3. Aaronson, S., Liu, J., Liu, Q., Zhandry, M., Zhang, R.: New approaches for quantum copy-protection. In: Advances in Cryptology—CRYPTO 2021. vol. 1, pp. 526–555 (2021). https://doi.org/10.1007/978-3-030-84242-0_19
4. Alagic, G., Majenz, C.: Quantum non-malleability and authentication. In: Advances in Cryptology—CRYPTO 2017. vol. 2, pp. 310–341 (2017). https://doi.org/10.1007/978-3-319-63715-0_11
5. Ananth, P., La Placa, R.L.: Secure software leasing. In: Advances in Cryptology—EUROCRYPT 2021. vol. 2, pp. 501–530 (2021). https://doi.org/10.1007/978-3-030-77886-6_17
6. Barnum, H., Crépeau, C., Gottesman, D., Smith, A., Tapp, A.: Authentication of quantum messages. In: 43rd Annual Symposium on Foundations of Computer Science—FOCS 2002. pp. 449–485 (2002). https://doi.org/10.1109/SFCS.2002.1181969
7. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: International Conference on Computers, Systems and Signal Processing. pp. 175–179 (1984)

8. Broadbent, A., Lord, S.: Uncloneable Quantum Encryption via Oracles. In: 15th Conference on the Theory of Quantum Computation, Communication and Cryptography—TQC 2020. pp. 4:1–4:22 (2020). https://doi.org/10.4230/LIPIcs.TQC.2020.4
9. Broadbent, A., Schaffner, C.: Quantum cryptography beyond quantum key distribution. Designs, Codes and Cryptography **78**(1), 351–382 (2016). https://doi.org/10.1007/s10623-015-0157-4
10. Cleve, R., Leung, D., Liu, L., Wang, C.: Near-linear constructions of exact unitary 2-designs. Quantum Information and Computation **16**(9-10), 721–756 (2016). https://doi.org/10.26421/QIC16.9-10-1
11. Coladangelo, A., Majenz, C., Poremba, A.: Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. arXiv preprint arXiv:2009.13865 (2020)
12. Dankert, C., Cleve, R., Emerson, J., Livine, E.: Exact and approximate unitary 2-designs and their application to fidelity estimation. Physical Review A **80**, 012304 (2009). https://doi.org/10.1103/PhysRevA.80.012304
13. Dieks, D.: Communication by EPR devices. Physics Letters A **92**(6), 271–272 (1982). https://doi.org/10.1016/0375-9601(82)90084-6
14. Dulek, Y., Speelman, F.: Quantum Ciphertext Authentication and Key Recycling with the Trap Code. In: 13th Conference on the Theory of Quantum Computation, Communication and Cryptography—TQC 2018. pp. 1:1–1:17 (2018). https://doi.org/10.4230/LIPIcs.TQC.2018.1
15. Dupuis, F., Nielsen, J.B., Salvail, L.: Actively secure two-party evaluation of any quantum operation. In: Advances in Cryptology—CRYPTO 2012. pp. 794–811 (2012). https://doi.org/10.1007/978-3-642-32009-5_46
16. Gao, J.: Quantum union bounds for sequential projective measurements. Physical Review A **92**(5), 052331 (2015). https://doi.org/10.1103/PhysRevA.92.052331
17. Garg, S., Yuen, H., Zhandry, M.: New security notions and feasibility results for authentication of quantum data. In: Advances in Cryptology—CRYPTO 2017. vol. 2, pp. 342–371 (2017). https://doi.org/10.1007/978-3-319-63715-0_12
18. Gottesman, D.: Uncloneable encryption. Quantum Information & Computation **3**(6), 581–602 (2003). https://doi.org/10.26421/QIC3.6-2
19. Kitagawa, F., Nishimaki, R., Yamakawa, T.: Secure software leasing from standard assumptions. arXiv preprint arXiv:2010.11186 (2020)
20. Mosca, M., Stebila, D.: Quantum coins. In: Error-Correcting Codes, Finite Geometries and Cryptography. pp. 35–47 (2010)
21. Naor, M., Reingold, O.: On the construction of pseudorandom permutations: Luby—rackoff revisited. Journal of Cryptology **12**(1), 29–66 (1999). https://doi.org/10.1007/PL00003817
22. Park, J.L.: The concept of transition in quantum mechanics. Foundations of Physics **1**(1), 23–33 (1970). https://doi.org/10.1007/BF00708652
23. Watrous, J.: The Theory of Quantum Information. Cambridge University Press, 1st edn. (2018)
24. Wiesner, S.: Conjugate coding. ACM SIGACT News **15**(1), 78–88 (1983). https://doi.org/10.1145/1008908.1008920
25. Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature **299**, 802–803 (1982). https://doi.org/10.1038/299802a0