

Public verification quantum money

Alex Bredariol Grilo



<https://abgrilo.github.io/viasm.html>



No-cloning theorem

Theorem: There is no quantum algorithm U such that for all $|\psi\rangle \in \mathbb{C}^2$, $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$.

Proof: Let us suppose that such a U exists. We have that

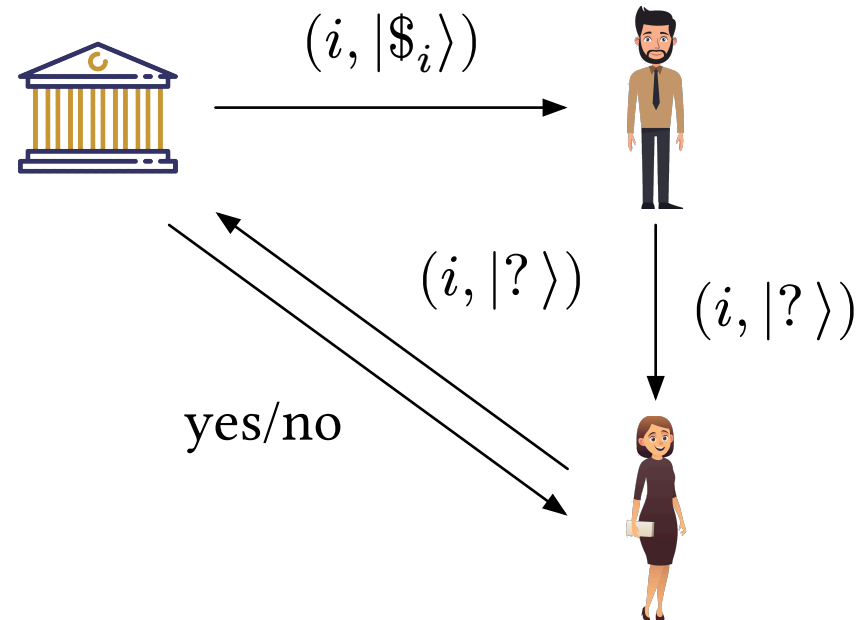
- $U|+\rangle|0\rangle = |+\rangle|+\rangle$
- $U|+\rangle|0\rangle = \frac{1}{\sqrt{2}}(U|0\rangle|0\rangle + U|1\rangle|0\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$

Contradiction!

No-cloning theorem

- Very powerful
- We can quantify how well we can do it (even with multiple copies)
 - Careful with the small print
 - Sub-families of quantum states
 - Information theoretical vs. computational statements
 - Variants of it can be used to construct many primitives: quantum money scheme, secure software leasing, ...

Private-verification quantum money schemes

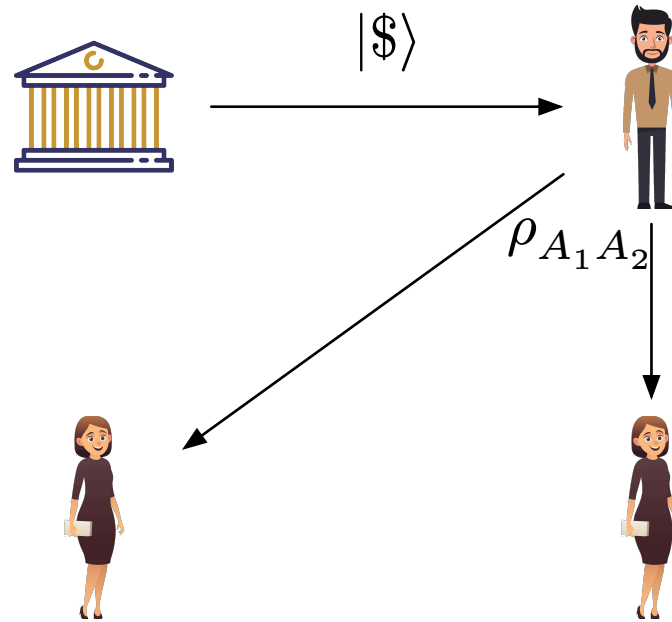


answers yes

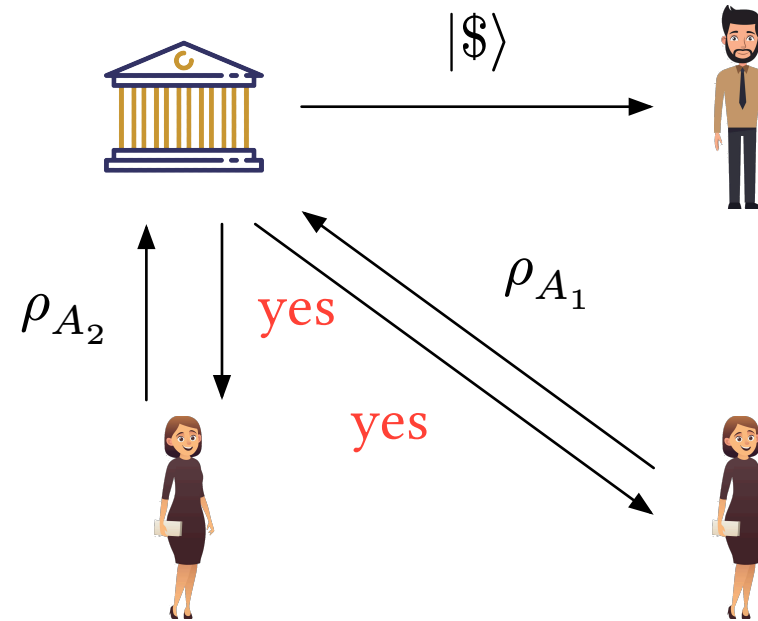
$$\Leftrightarrow |?\rangle = |\$_i\rangle$$

For simplicity, serial number i will be omitted in the following slides.

Private-verification quantum money schemes - security

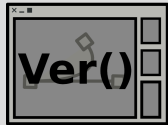
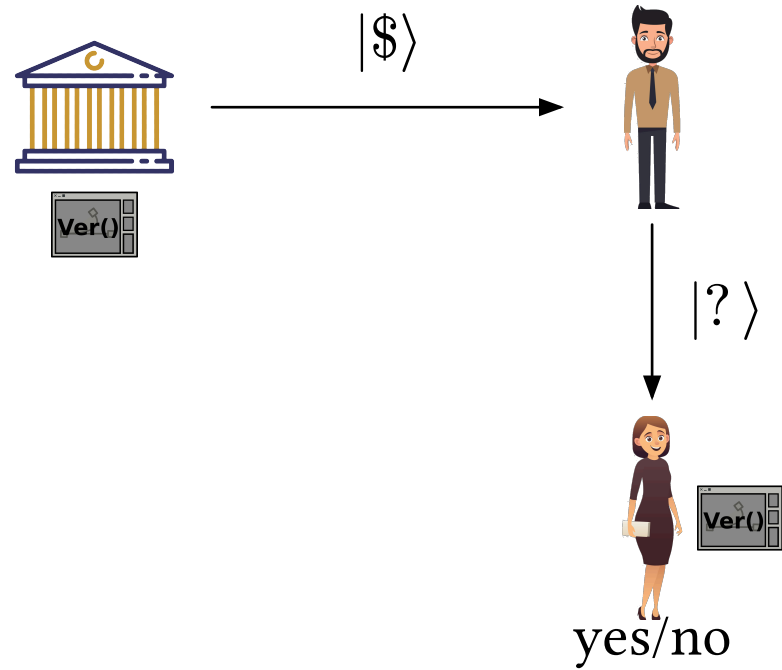


Private-verification quantum money schemes - security



Security (informal): Bob cannot create two states such that both pass the test.

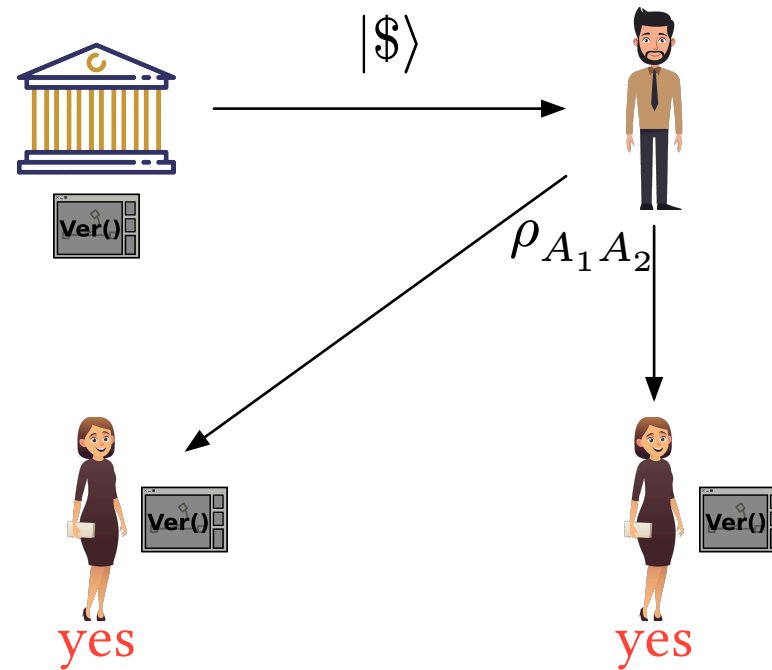
Public-verification quantum money schemes



that Alice runs answers yes

$$\Leftrightarrow |\? \rangle = |\$ \rangle$$

Public-verification quantum money schemes - security



Security (informal): Bob cannot create two states such that both pass the test.

Definition of PV quantum money

- Key Generation: $\text{KeyGen}(1^\lambda) \mapsto (pk, sk)$
- Minting: $\text{Mint}(sk) \mapsto \rho$
- Verification: $\text{Ver}_{pk}(\rho) \mapsto \{0, 1\}$

Correctness: $\Pr[\text{Ver}_{pk}(\text{Mint}(sk)) = 1] = 1$

Security: $\forall$ adversaries $\mathcal{A}$:

$$\Pr[\text{Ver}_{pk} \otimes \text{Ver}_{pk}(\mathcal{A}(pk, \text{Mint}(sk))) = (1, 1)] \leq \text{negl}(\lambda)$$

(Im)possibility of information-theoretical quantum money

Theorem (Wiesner'68). There is a private-verification quantum money scheme that is information theoretically secure.

Theorem: There is no information-theoretical PV quantum money scheme.

Proof sketch: Let $(\text{KeyGen}, \text{Mint}, \text{Ver})$ be an candidate PV quantum money scheme.

The following (exponential-time) adversary can break it:

1. For all possible sk' :
 - 1.1 If $\text{Ver}_{pk}(\text{Mint}(sk')) = 1$, output $\text{Mint}(sk') \otimes \text{Mint}(sk')$.

Goal: Construct PV quantum money that is safe against poly-time adversaries

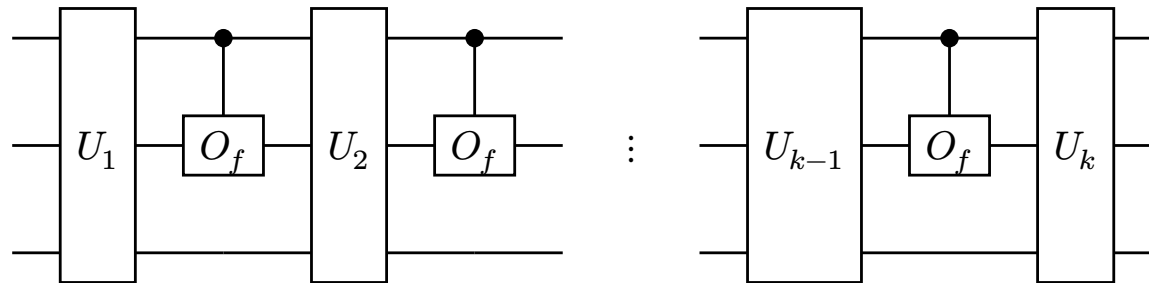
Still a very hard task. Let's try something simpler.

Oracles

Definition: Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$.

$$O_f|x\rangle|b\rangle = |x\rangle|b \oplus f(x)\rangle$$

C^{O_f} - Quantum circuit C with O_f oracle access:



U_i 's are fully known but only black-box access to O_f .



We can only learn f from the outcomes of oracle calls.

PV quantum money with oracles

Simpler task:

Instead of outputting pk , KeyGen will output sk which is still used for minting, and an oracle O_{sk} which is used for verification.

Correctness (with oracles): $\Pr[\text{Ver}^{O_{sk}}(\text{Mint}(sk)) = 1] = 1$

Security (with oracles): $\forall$ QPT $\mathcal{A}$:

$$\Pr[\text{Ver}^{O_{sk}} \otimes \text{Ver}^{O_{sk}}(\mathcal{A}^{O_{sk}}(\text{Mint}(sk))) = (1, 1)] \leq \text{negl}(\lambda)$$

Goal: Construct PV quantum money scheme with oracles (Aaronson-Christiano'12)

Subspace states

Definitions: Let us fix a subspace $S \subseteq \mathbb{F}_2^n$ of dimension $\frac{n}{2}$

- $|S\rangle = \frac{1}{\sqrt{2^{\frac{n}{2}}}} \sum_{\{x \in S\}} |x\rangle$
- $S^\perp = \{y \in \mathbb{F}_2^n : y \perp S\}$
 $S^\perp$ is also a subspace of $\mathbb{F}_2^n$ of dimension $\frac{n}{2}$
- $O_S |x\rangle |b\rangle = |x\rangle |b \oplus \mathbb{1}_{x \in S}\rangle$

Subspace states - important property

Theorem: $H^{\otimes n}|S\rangle = |S^\perp\rangle$

Proof:

$$\begin{aligned} H^{\otimes n}|S\rangle &= \frac{1}{\sqrt{2^{\frac{n}{2}}}} \sum_{x \in S} H^{\otimes n}|x\rangle \\ &= \frac{1}{\sqrt{2^{\frac{n}{2}}} 2^n} \sum_{x \in S} \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y} |y\rangle \quad (\text{exercise}) \\ &= \frac{1}{\sqrt{2^{\frac{n}{2}}} 2^n} \sum_{y \in \mathbb{F}_2^n} \left(\sum_{x \in S} (-1)^{x \cdot y} \right) |y\rangle \end{aligned}$$

If $y \in S^\perp$, then $x \cdot y = 0$ for all $x \in S$, so its amplitude is $\frac{1}{\sqrt{2^{\frac{n}{2}}}}$.

If $y \notin S^\perp$, then its amplitude is zero (**exercise**).

Therefore $H^{\otimes n}|S\rangle = |S^\perp\rangle$

PV quantum money from subspace states

- $\text{KeyGen}(1^\lambda)$:
 1. Sample $S \subseteq \mathbb{F}_2^n$ of dimension $\frac{n}{2}$
 2. Set $sk = S$ and $O_{sk} = (O_S, O_{S^\perp})$
- $\text{Mint}(S)$:
 1. Output $|S\rangle$
- $\text{Ver}^{O_S, O_{S^\perp}}(\rho) \mapsto \{0, 1\}$
 1. Run O_S on $\rho \otimes |0\rangle\langle 0|$.
 2. Measure the last qubit and output 0 if the outcome is 0.
 3. Apply $H^{\otimes n}$ onto the first register and let ρ' be the resulting state.
 4. Run $O_{S^\perp}$ on $\rho' \otimes |0\rangle\langle 0|$.
 5. Measure the last qubit and output 0 if the outcome is 0.
 6. Otherwise, output 1.

Correctness

Theorem: $\text{Ver}^{O_S, O_{S^\perp}}(|S\rangle\langle S|) = 1$

Proof:

1. $O_S|S\rangle|0\rangle = \frac{1}{\sqrt{2^{\frac{n}{2}}}} \sum_{x \in S} O_S|x\rangle|0\rangle = \frac{1}{\sqrt{2^{\frac{n}{2}}}} \sum_{x \in S} |x\rangle|1\rangle = |S\rangle|1\rangle$
 $\Rightarrow$ first measurement never rejects.
2. $H^{\otimes n}|S\rangle = |S^\perp\rangle$
3. $O_{S^\perp}|S^\perp\rangle|0\rangle = |S^\perp\rangle|1\rangle \Rightarrow$ second measurement never rejects.

Remark: $\text{Ver}^{O_S, O_{S^\perp}} \equiv$ measuring according to $\{|S\rangle\langle S|, I - |S\rangle\langle S|\}$

Security

Goal: Show that in the black-box setting, any adversary needs exponentially many oracle queries.

Potential attacker:

A perfect counterfeiter must transform

$$|S\rangle \mapsto |S\rangle|S\rangle \quad \text{and} \quad |S'\rangle \mapsto |S'\rangle|S'\rangle$$

Notice that $\langle S|^{\otimes 2}|S'\rangle^{\otimes 2} = \langle S|S'\rangle^2$, which can be much smaller than $\langle S|S'\rangle$.



Cloning forces the inner product to decrease.

Proof strategy

For $(O_S, O_{S^\perp})$, let $|\Psi_S^t\rangle$ be state after the t th oracle call.

For two chosen subspaces S, S' from some special distribution D , we can define the progress measure

$$p_t = E_{(S, S') \sim D} [\langle \Psi_S^t | \Psi_{S'}^t \rangle]$$

We want:

1. p_0 is large.
2. p_T is small.
3. A single query can decrease p_t only slightly.

From progress to a query lower bound

Theorem. Suppose initially $|\langle \Psi_S^0 | \Psi_{S'}^0 \rangle| \geq c$, while successful counterfeiting requires $|\langle \Psi_S^T | \Psi_{S'}^T \rangle| \leq d$.

If every query changes the progress by at most δ , then the number of queries to correctly copy the states are

$$T = \Omega\left(\frac{c-d}{\delta}\right)$$

Proof.

$$p_T \geq p_0 - T\delta \text{ and } p_0 - p_T = c - d$$

$\Downarrow$

$$T = \Omega\left(\frac{c-d}{\delta}\right)$$

Key step

1. p_0 is large.
2. p_T is small.
3. **A single query can decrease p_t only slightly.**

Lemma: $p_t \geq p_{t-1} - O(2^{-\frac{n}{4}})$.

Proof: Complicated: highly based on quantum query complexity.

Theorem (Aaronson-Christiano). Any adversary that breaks the subspace quantum money must make $O(2^{\frac{n}{4}})$ queries to the oracle $\Rightarrow$ No QPT adversary can break the scheme.

We have a secure PV quantum-money scheme 🎉 but only with oracles 😭

Goal: Construct PV quantum-money scheme in the plain model from the subspace quantum money oracle scheme (Zhandry'19)

From oracles to the plain model - Obfuscation

Intuition: A program is obfuscated if $\text{Obf}(C)$ has the same functionality of C , but it does not tell us anything from the implementation of C .

Codify Formatter

JavaScript Obfuscator

Input

Sample

```
1- for (var line = 1; line < 10; line++) {  
2-   for (var i = 1; i < 29; i++) {  
3-     var s = Math.floor(2 * Math.random() % 2) ? "  
4-       *\" : \"*\";  
5-     document.write(s)  
6-   }  
7-   document.writeln("<br>")  
}
```

Obfuscator JS

Beautify JS

Minify JS

Run/View HTML

Output

```
1 var _0x2f9796=_0x293d;function _0x293d(_0xa07fa9  
,_0xc54dcd){var _0x1448cf=_0x1448();return _0x293d  
=function(_0x293d0b,_0x595be3){_0x293d0b=_0x293d0b  
-0x1cd;var _0x135663=_0x1448cf[_0x293d0b];return  
_0x135663;},_0x293d(_0xa07fa9,_0xc54dcd);}(function  
(_0x2bee25,_0x21e2c3){var _0x106517=_0x293d  
,_0x9308a7=_0x2bee25();while(![]){try{var  
_0x562ac7=-parseInt(_0x106517(0x1ce))/0x1+parseInt  
(_0x106517(0x1da))/0x2+-parseInt(_0x106517(0x1d1  
))/0x3*(parseInt(_0x106517(0x1d3))/0x4)+-parseInt  
(_0x106517(0x1d2))/0x5+-parseInt(_0x106517(0x1d4  
))/0x6+-parseInt(_0x106517(0x1d7))/0x7*(-parseInt  
(_0x106517(0x1d6))/0x8)+parseInt(_0x106517(0x1d9  
))/0x9;if(_0x562ac7===_0x21e2c3)break;else  
_0x9308a7['push'](_0x9308a7['shift']());}catch  
(_0x36640a){_0x9308a7['push'](_0x9308a7['shift']  
());}}(_0x1448,0xe3966));for(var line=0x1;line<0xa  
;line++){for(var i=0x1;i<0x1d;i++){var s  
=Math[_0x2f9796(0x1d8)](0x2*Math[_0x2f9796(0x1d5)]  
()%0x2)>'/'?:'\';document[_0x2f9796(0x1d0)](s  
);}document[_0x2f9796(0x1cd)](_0x2f9796(0x1cf  
));}function _0x1448(){var _0x4c6f57=['7rZrRVP'  
, 'floor', '24411789nEYSPv', '115888JawEzP', 'writeln'  
, '697895uXCkzB', '<br>', 'write', '6ok1Vvv'  
, '346160HmKobD', '2674648WUTUtn', '2792574biNBap'  
, 'random', '5853704KLyHw'];_0x1448=function  
(){return _0x4c6f57;};return _0x1448();}
```

From oracles to the plain model - Attempt 1

Virtual Black-Box obfuscation (VBB): A VBB obfuscator Obf for a circuit C produces $\text{Obf}(C)$ with the same functionality as C such that for any QPT adversary $\mathcal{A}$, there exists a QPT simulator $\mathcal{S}$ such that:

$$\left| \Pr[\mathcal{A}(\text{Obf}(C, 1^\lambda)) = 1] - \Pr[\mathcal{S}^{C(1^\lambda)} = 1] \right| \leq \text{negl}(\lambda).$$

Theorem. If VBB obfuscation for O_S exists, then there exists a PV quantum money scheme.

Theorem. VBB is impossible for general circuits.

Possible that VBB obfuscation exists for the specific circuits O_S and $O_{S^\perp}$, but there is no indication that this is the case.

From oracles to the plain model - Attempt 2

subspace-hiding Obfuscation (shO): A subspace-hiding obfuscator shO is a PPT algorithm such that $\text{shO}(O_S)$ has the same functionality as O_S and the following holds.

Let us consider the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

1. $\mathcal{A}$ sends a subspace S_0 of dimension d_0 to $\mathcal{C}$.
2. $\mathcal{C}$ chooses a random subspace $S_1 \subseteq F_2^n$ of dimension d_1 such that $S_0 \subseteq S_1$.
3. $\mathcal{C}$ picks b uniformly at random and gives $\text{shO}(S_b)$ to $\mathcal{A}$.
4. $\mathcal{A}$ makes a guess b' .
5. $\mathcal{A}$ wins if $b = b'$.

shO is secure if all QPT adversaries have negligible advantage over a random guess.

PV quantum money from shO

Theorem. Let shO be a secure subspace-hiding obfuscator for $d_0 = \frac{n}{2}$ and some d_1 such that both $|F|^{n-d_1}$ and $|F|^{d_1-\frac{n}{2}}$ are exponentially large.

If we instantiate the oracle from the previous scheme with shO, we have a secure quantum money scheme in the plain model.

Proof strategy: Assume that, given a valid banknote, an adversary can produce two banknotes that pass verification with probability ε .

We construct hybrids $H_0 \rightarrow H_1 \rightarrow H_2 \rightarrow H_3 \rightarrow H_4 \rightarrow H_5$ such that:

- H_0 is the real security experiment;
- Consecutive hybrids are computationally indistinguishable; and
- We can prove that H_5 is secure.

Hybrid H_0 - real security experiment

1. $\mathcal{C}$ chooses a random subspace S of dimension $\frac{n}{2}$.
2. $\mathcal{C}$ prepares the valid banknote $|S\rangle$.
3. $\mathcal{C}$ gives $|S\rangle$ and $(\text{shO}(O_S), \text{shO}(O_{S^\perp}))$ to $\mathcal{A}$.
4. $\mathcal{A}$ gives a quantum state $\rho_{A,B}$ to $\mathcal{C}$.
5. $\mathcal{C}$ runs the verification algorithm on both registers with $(\text{shO}(O_S), \text{shO}(O_{S^\perp}))$ to implement the oracles.
6. $\mathcal{A}$ wins iff both verification procedures pass.

Assumption: $\Pr[\mathcal{A} \text{ wins in } H_0] = \varepsilon$.

Hybrid H_1 : projective verification

Consider the following modification from H_0 :

5. Instead of running the verification algorithm, $\mathcal{C}$ measures $\rho_{A,B}$ according to $\{\Pi_0 = (|S\rangle\langle S|)^{\otimes 2}, \Pi_1 = I - \Pi_0\}$
6. $\mathcal{A}$ wins iff the outcome is 0

Lemma: $\Pr[\mathcal{A} \text{ wins in } H_1] = \varepsilon.$

Proof: The verification algorithm is equivalent to the projective measurement.

Hybrid H_2 : enlarge S

Let $S \subseteq T$ for a random subspace T of dimension d_1 .

Consider the following modification from H_1 :

3. $\mathcal{C}$ gives $|S\rangle$ and $(\text{shO}(O_T), \text{shO}(O_{S^\perp}))$ to $\mathcal{A}$.

Lemma: $\Pr[\mathcal{A} \text{ wins in } H_2] = \varepsilon + \text{negl}(n)$.

Proof: Subspace-hiding property of shO .

Hybrid H_3 : enlarge $S^\perp$

Let $S^\perp \subseteq T'$ for a random subspace T' of dimension d_1

Consider the following modification from H_2 :

3. $\mathcal{C}$ gives $|S\rangle$ and $(\text{shO}(O_T), \text{shO}(O_{T'}))$ to $\mathcal{A}$.

Lemma: $\Pr[\mathcal{A} \text{ wins in } H_3] = \varepsilon + \text{negl}(n)$.

Proof: Subspace-hiding property of shO .

Hybrid H_4

We now change the order for sampling S, T, T' :

1. $\mathcal{C}$ picks a random subspace T of dimension d_1
2. $\mathcal{C}$ picks a random subspace T' of dimension d_1 such that $T^\perp \subseteq T'$
3. $\mathcal{C}$ picks a random S such that $(T')^\perp \subseteq S \subseteq T$
4. $\mathcal{C}$ gives $|S\rangle$ and $(\text{shO}(O_T), \text{shO}(O_{T'}))$ to $\mathcal{A}$.

Lemma: $\Pr[\mathcal{A} \text{ wins in } H_4] = \Pr[\mathcal{A} \text{ wins in } H_3] = \varepsilon + \text{negl}(n)$.

Proof: New sampling procedure gives the same distribution.

Hybrid H_5 : Fixing T, T'

$$\Pr_{\tilde{T}, \tilde{T}', S} [\mathcal{A} \text{ wins in } H_4] = \varepsilon + \text{negl}(n)$$

$\Downarrow$

$$\exists \tilde{T}, \tilde{T}' \text{ such that } \Pr_S [\mathcal{A} \text{ wins in } H_4 \mid \tilde{T}, \tilde{T}'] \geq \varepsilon + \text{negl}(n)$$

H_5 : Fix (T, T') to be $(\tilde{T}, \tilde{T}')$.

Since $\mathcal{A}$ knows them, so $\mathcal{C}$ doesn't need to send $(\text{shO}(O_T), \text{shO}(O_{T'}))$ to $\mathcal{A}$.

Lemma. $\Pr[\mathcal{A} \text{ wins in } H_5] \geq \Pr[\mathcal{A} \text{ wins in } H_4] = \varepsilon + \text{negl}(n)$.

Bounding the probabilities

Lemma:

The probability that an adversary can clone $|S\rangle$ for a random $\tilde{T}'^\dagger \subseteq S \subseteq \tilde{T}$ is $\text{negl}(n) \Rightarrow \Pr[\mathcal{A} \text{ wins in } H_5] \leq \text{negl}(n)$.

Proof: Stronger version of no-cloning theorem.

$$\Pr[\mathcal{A} \text{ wins in } H_5] \geq \Pr[\mathcal{A} \text{ wins in } H_0] + \text{negl}(n) \quad \text{and} \quad \Pr[\mathcal{A} \text{ wins in } H_5] = \text{negl}(n)$$



$$\Pr[\mathcal{A} \text{ wins in } H_0] = \text{negl}(n)$$

Unless $\mathcal{A}$ breaks shO, they cannot produce two valid banknotes with non-negligible probability 🎉 **Do shO schemes exist?** 😬

Indistinguishability obfuscation

indistinguishability Obfuscation (iO): An iO obfuscator iO is a PPT algorithm such that for any two circuits C_1, C_2 of size $\text{poly}(\lambda)$ with identical input-output behavior, and for any QPT adversary $\mathcal{A}$:

$$|\Pr[\mathcal{A}(\text{iO}(C_1)) = 1] - \Pr[\mathcal{A}(\text{iO}(C_2)) = 1]| \leq \text{negl}(\lambda)$$

There exist candidates for iO constructions, but status is still unclear (specially for post-quantum security). At least no impossibility result is known.

From iO to shO

Theorem. If injective one-way functions exist, then any indistinguishability obfuscator is also a subspace-hiding obfuscator for dimensions d_0, d_1 , as long as 2^{d_0} and 2^{n-d_1} are exponential.

Proof strategy: We will consider the case where $d_1 = d_0 + 1$, and the general case follows by induction.

Let $\mathcal{A}$ be an adversary against subspace hiding.

We construct the hybrids $H_0 \rightarrow H_1 \rightarrow H_2 \rightarrow H_3 \rightarrow H_4$, where H_0 is the $b = 0$ experiment and H_4 is the $b = 1$ experiment.

H_0 and H_1

H_0 :

1. $\mathcal{A}$ submits a d_0 -dimensional subspace S_0 .
2. $\mathcal{C}$ gives $\text{iO}(O_{S_0})$ to $\mathcal{A}$.
3. $\mathcal{A}$ guesses b' .

H_1 :

Let B be a basis for $S_0^\perp$. Let $\hat{P}$ be the iO of the program Z that always outputs 0.

2. $\mathcal{C}$ gives the adversary $\text{iO}(Q)$, where $Q(x) = \begin{cases} 1 & \text{if } Bx=0 \\ 1 & \text{if } \hat{P}(Bx)=1. \\ 0 & \text{otherwise.} \end{cases}$

Lemma: $\Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_0] - \Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_1] \leq \text{negl}(n)$.

Proof: $Q(x) = 1 \Leftrightarrow x \in S_0$. The result follows from the security of iO.

H_2

Pick a random x^* and let $P_y(x) = \begin{cases} 1 & \text{if } \text{OWF}(x) = \text{OWF}(x^*) \\ 0 & \text{otherwise} \end{cases}$

2. $\mathcal{C}$ gives the adversary $\text{iO}(Q')$, where $Q'(x) = \begin{cases} 1 & Bx=0 \\ 1 & P_y(Bx)=1. \\ 0 & \text{otherwise.} \end{cases}$

Lemma: $\Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_1] - \Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_2] \leq \text{negl}(n).$

Proof: The programs Z and P_y differ only on the input x^* . We can show that distinguishing their obfuscations would allow us to find x^* and invert the one-way function.

H_3

2. $\mathcal{C}$ picks a random x^* , let $S_1 = \{y : By \in \{0, x^*\}\}$, and $\mathcal{C}$ gives $\text{iO}(O_{S_1})$ to $\mathcal{A}$.

Lemma: $\Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_2] = \Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_3]$.

Proof:

$$Q'(x) = 1 \Leftrightarrow Bx \in \{0, x^*\} \Leftrightarrow x \in \{y : By \in \{0, x^*\}\}$$

H_4

2. $\mathcal{C}$ picks a random $x^* \neq 0^n$ and let $S_1 = \{y : By \in \{0, x^*\}\}$, and $\mathcal{C}$ gives $\text{iO}(O_{S_1})$ to $\mathcal{A}$.

Now S_1 is a uniformly random d_1 -dimensional subspace containing $S_0 \Rightarrow H_4$ is exactly the $b = 1$ experiment.

Lemma: $\Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_3] - \Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_4] \leq \text{negl}(n)$.

Proof: The probability that a random x^* equals 0 is

$$\frac{1}{2^{n-d_0}} = \frac{1}{2^{n-d_1+1}} = \text{negl}(n).$$

Wrapping up

Lemma. $\Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_0] - \Pr[\mathcal{A} \text{ guesses } 0 \text{ in } H_4] \leq \text{negl}(n).$

Thus the obfuscation of S_0 is computationally indistinguishable from the obfuscation of a random d_1 -dimensional subspace containing S_0 .

Hence iO and injective one-way functions give us a secure subspace-hiding obfuscator.

Open Question

We have seen PV quantum money from iO. But we don't have any construction for iO from post-quantum assumptions.

Other constructions from ad-hoc assumptions.

Open question: Can we build PV quantum money from standard cryptographic assumptions?

Related primitives

- Unclonable functionalities
 - Uncloneable encryption
 - Uncloneable programs
 - One-time programs
 - Uncloneable proofs
- Certified deletion
- One-shot signatures
- ...

Questions?