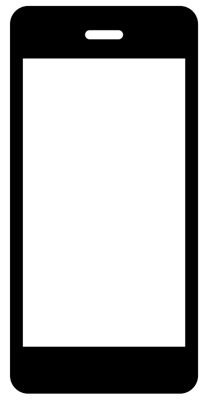
How to construct Quantum FHE, Generically

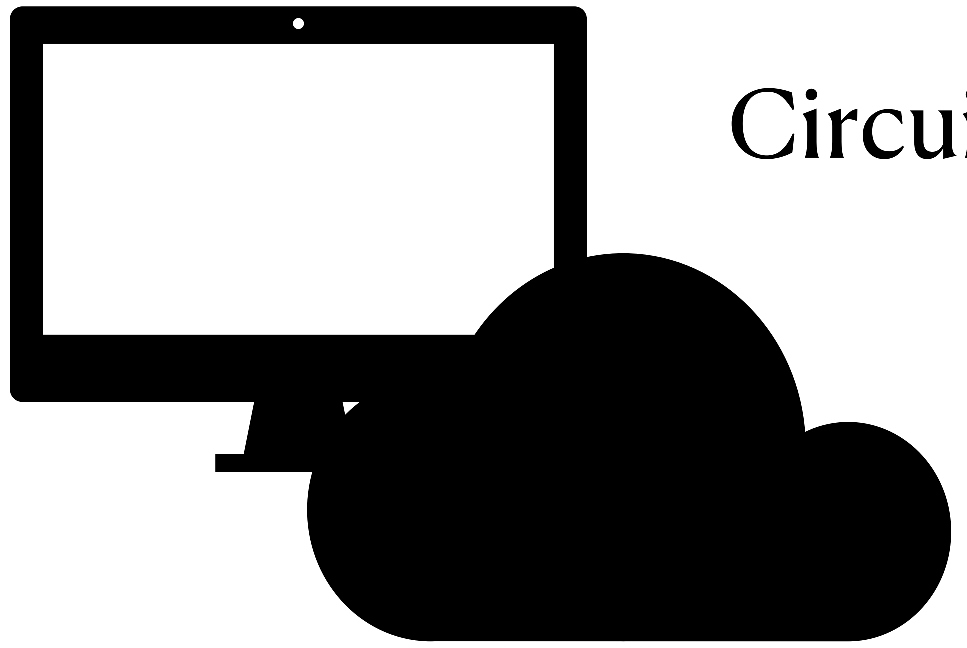
Aparna Gupte
MIT

Vinod Vaikuntanathan
MIT

Classical Fully Homomorphic Encryption (FHE)

Input x



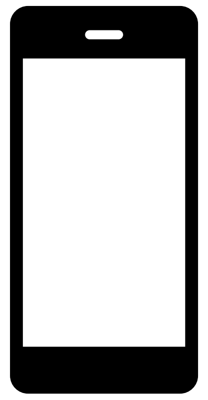


Circuit C

Server should not learn anything about x

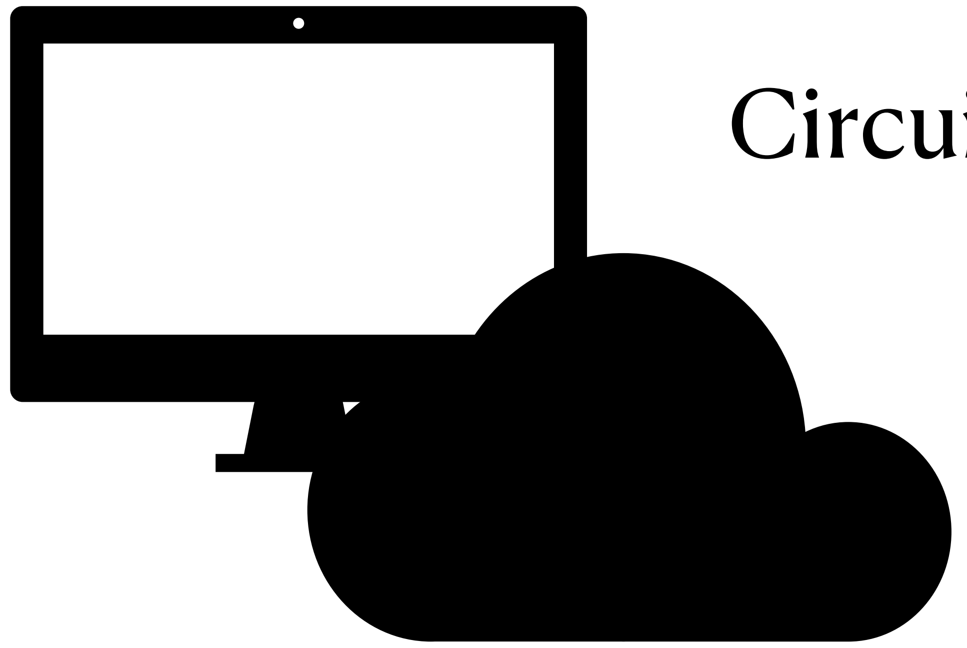
Classical Fully Homomorphic Encryption (FHE)

Input x



c, evk

$\longrightarrow$



Circuit C

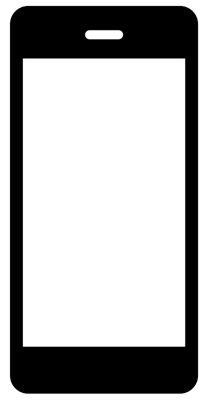
$pk, sk, evk \leftarrow \mathrm{KeyGen}$

$c \leftarrow \mathrm{Enc}_{pk}(x)$

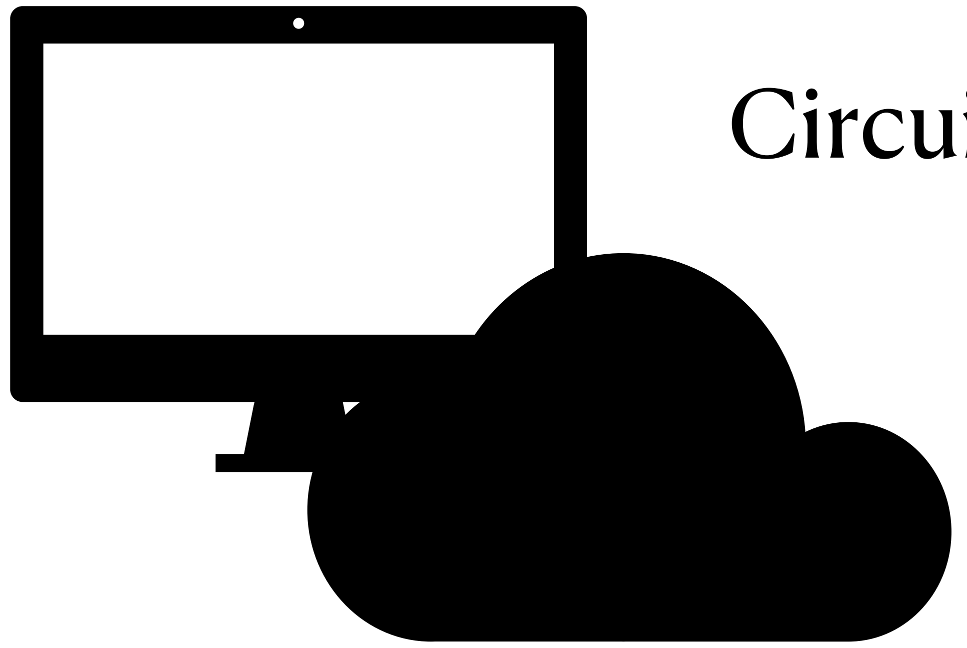
Server should not learn anything about x

Classical Fully Homomorphic Encryption (FHE)

Input x



c, evk $\longrightarrow$



Circuit C

$pk, sk, evk \leftarrow \mathrm{KeyGen}$

$c \leftarrow \mathrm{Enc}_{pk}(x)$

$C(x) \leftarrow \mathrm{Dec}_{sk}(c')$

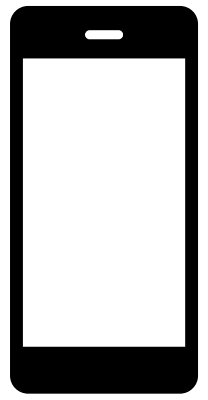
$\longleftarrow$ c'

$c' \leftarrow \mathrm{Eval}(C, c, evk)$

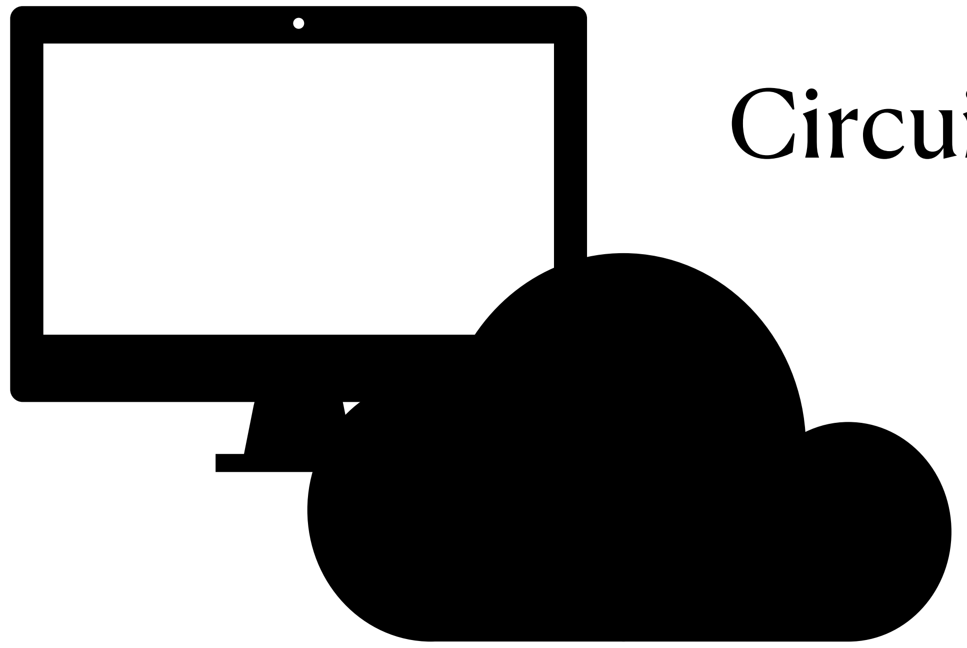
Server should not learn anything about x

Classical Fully Homomorphic Encryption (FHE)

Input x



c, evk $\longrightarrow$



Circuit C

$pk, sk, evk \leftarrow \text{KeyGen}$

$c' \longleftarrow$

$c \leftarrow \text{Enc}_{pk}(x)$

$c' \leftarrow \text{Eval}(C, c, evk)$

$C(x) \leftarrow \text{Dec}_{sk}(c')$

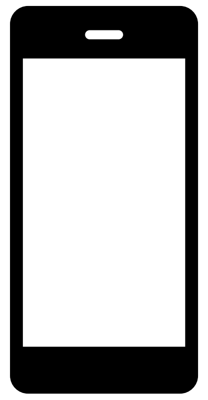
Server should not learn anything about x

Client efficiency:

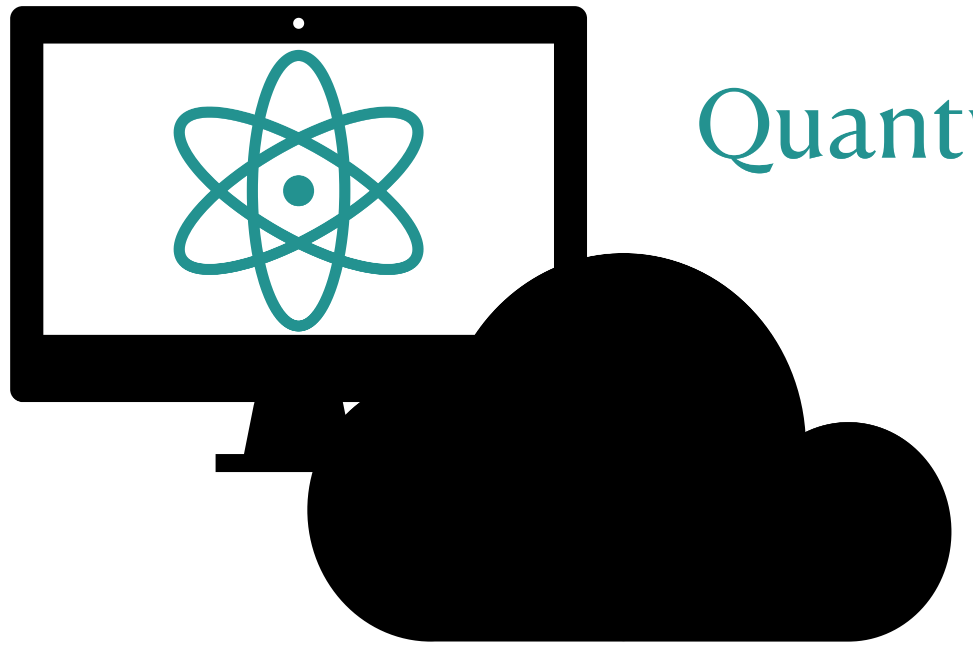
- Decryption should be more efficient than computing C

Quantum Fully Homomorphic Encryption (QFHE)

Input x



c, evk $\longrightarrow$



Quantum circuit Q

$pk, sk, evk \leftarrow \text{KeyGen}$

$c \leftarrow \text{Enc}_{pk}(x)$

c' $\longleftarrow$

$c' \leftarrow \text{Eval}\,(Q, c, evk)$

$Q(x) \leftarrow \text{Dec}_{sk}(c')$

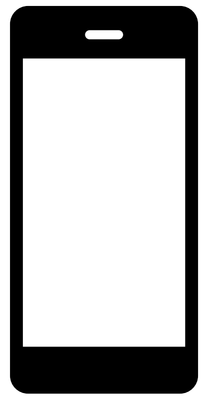
Server should not learn anything about x

Client efficiency:

- Decryption should be more efficient than computing Q

Quantum Fully Homomorphic Encryption (QFHE)

Input x



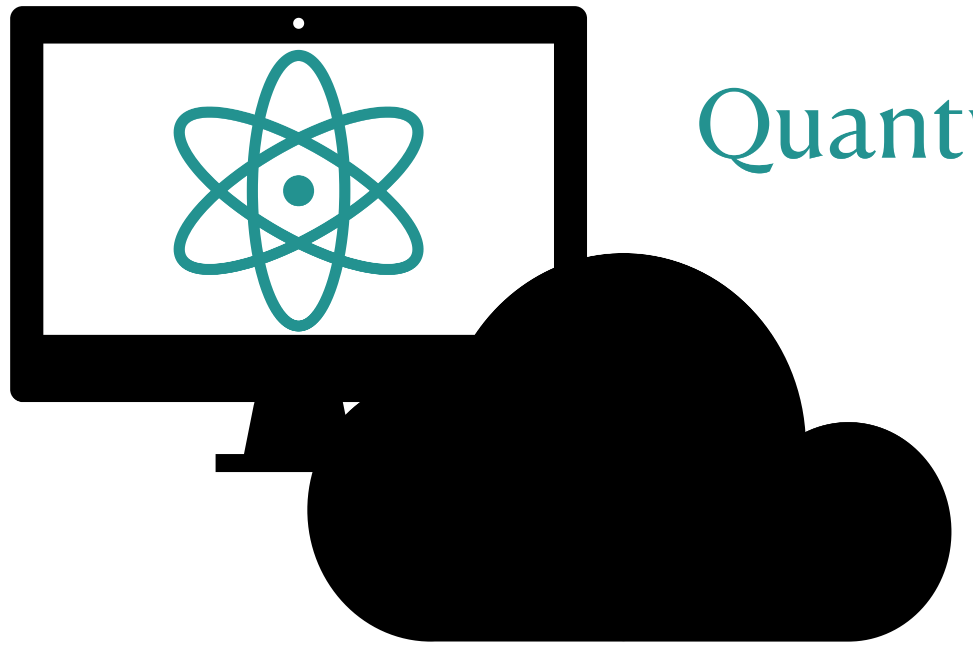
$pk, sk, evk \leftarrow \text{KeyGen}$

$c \leftarrow \text{Enc}_{pk}(x)$

$Q(x) \leftarrow \text{Dec}_{sk}(c')$

$\xrightarrow{c,\ evk}$

$\xleftarrow{c'}$



Quantum circuit Q

$c' \leftarrow \text{Eval}\,(Q, c, evk)$

Server should not learn anything about x

Client efficiency:

- Decryption should be more efficient than computing Q
- **Client should be classical!**

What was known

What was known

	Quantum/Classical client	Assumptions needed
BJ15, DSS16	Quantum client	Any post-quantum (pq.) classical FHE*

* with decryption circuit in NC1

What was known

	Quantum/Classical client	**Assumptions needed**
BJ15, DSS16	Quantum client 🙁	Any post-quantum (pq.) classical FHE*

* with decryption circuit in NC1

What was known

	Quantum/Classical client	Assumptions needed
BJ15, DSS16	Quantum client 🙁	Any post-quantum (pq.) classical FHE* 🙂

* with decryption circuit in NC1

What was known

	Quantum/Classical client	**Assumptions needed**
BJ15, DSS16	Quantum client 🙁	Any post-quantum (pq.) classical FHE* 🙂
Mah18a, Bra18	Classical client	Learning With Errors (LWE)

* with decryption circuit in NC1

What was known

	Quantum/Classical client	Assumptions needed
BJ15, DSS16	Quantum client 🙁	Any post-quantum (pq.) classical FHE* 🙂
Mah18a, Bra18	Classical client 🙂	Learning With Errors (LWE)

* with decryption circuit in NC1

What was known

	Quantum/Classical client	Assumptions needed
BJ15, DSS16	Quantum client 🙁	Any post-quantum (pq.) classical FHE* 🙂
Mah18a, Bra18	Classical client 🙂	Learning With Errors (LWE) 🙁

* with decryption circuit in NC1

Does (classical-client) QFHE really need LWE?

Does (classical-client) QFHE really need LWE?

Why should we diversify assumptions?

Does (classical-client) QFHE really need LWE?

Quantum Algorithms for Lattice Problems

Yilei Chen*

April 18, 2024

Why should we diversify assumptions?

- Do not want all QFHE eggs in the LWE basket. What if we break LWE?

Does (classical-client) QFHE really need LWE?

Quantum Algorithms for Lattice Problems

Yilei Chen*

April 18, 2024

Why should we diversify assumptions?

- Do not want all QFHE eggs in the LWE basket. What if we break LWE?
- Different properties useful in different contexts: different efficiency profiles

Does (classical-client) QFHE really need LWE?

Quantum Algorithms for Lattice Problems

Yilei Chen*

April 18, 2024

Why should we diversify assumptions?

- Do not want all QFHE eggs in the LWE basket. What if we break LWE?
- Different properties useful in different contexts: different efficiency profiles

Dream Theorem. Any post-quantum classical FHE $\Longrightarrow$ QFHE.

Our Results (QFHE)

Theorem 1 [G-Vaikuntanathan]

pq. Classical FHE* + pq. Dual-mode Trapdoor Functions $\implies$ QFHE

* with decryption circuit in NC1

Our Results (QFHE)

Theorem 1 [G-Vaikuntanathan]

pq. Classical FHE* + pq. Dual-mode Trapdoor Functions $\Longrightarrow$ QFHE

$\Uparrow$

- LWE [BV11, BV14]
- pq. IO + pq. re-randomizable encryption [CLTV14]
- pq. IO + Group actions [CLTV14, Wichs'24]

* with decryption circuit in NC1

Our Results (QFHE)

Theorem 1 [G-Vaikuntanathan]

pq. Classical FHE* + pq. Dual-mode Trapdoor Functions $\Longrightarrow$ QFHE

- LWE [BV11, BV14]
- pq. IO + pq. re-randomizable encryption [CLTV14]
- pq. IO + Group actions [CLTV14, Wichs'24]

- LWE [Mah18b]
- Group actions **[GV24, Theorem 2]**

Build on work of Alamati, Malavolta and Rahimi [AMR22]

* with decryption circuit in NC1

Our Results (QFHE)

Theorem 1 [G-Vaikuntanathan]

pq. Classical FHE* + pq. Dual-mode Trapdoor Functions $\implies$ QFHE

- LWE [BV11, BV14]
- pq. IO + pq. re-randomizable encryption [CLTV14]
- pq. IO + Group actions [CLTV14, Wichs'24]

- LWE [Mah18b]
- Group actions **[GV24, Theorem 2]**

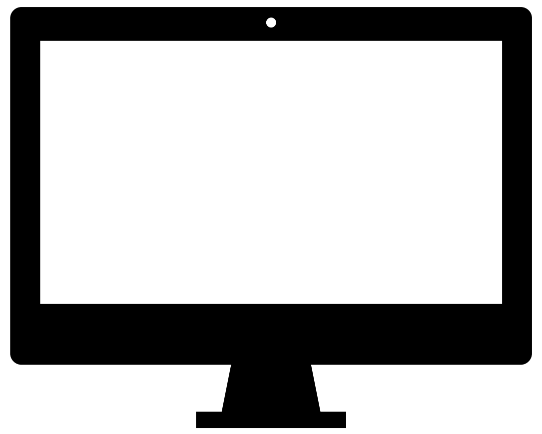
Build on work of Alamati, Malavolta and Rahimi [AMR22]

Corollary [GV24]. pq. IO + Group actions $\implies$ QFHE

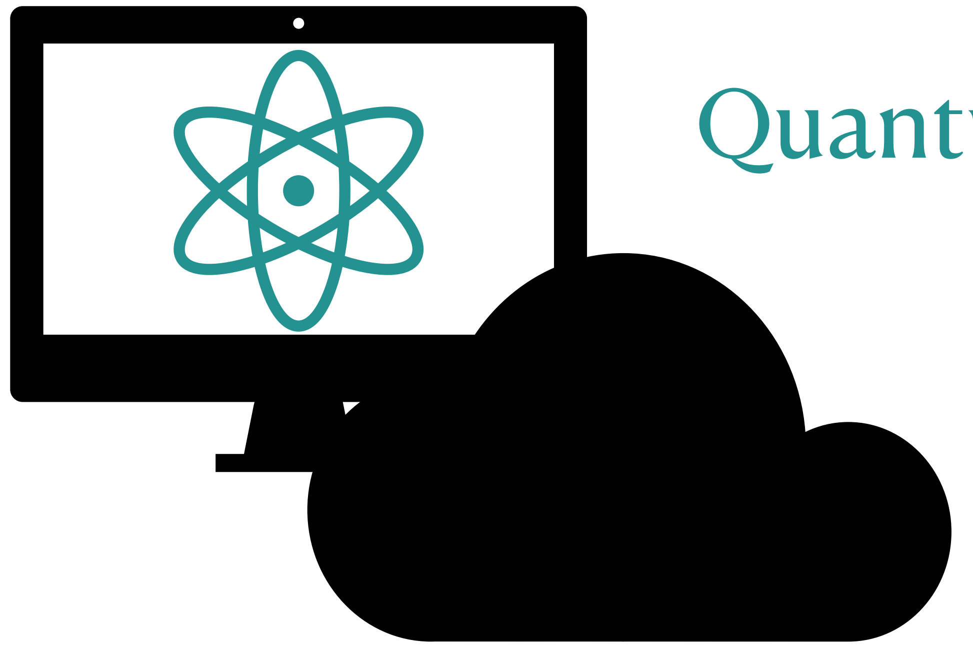
* with decryption circuit in NC1

Non-compact QFHE

Input x



$c, evk \longrightarrow$



Quantum circuit Q

$pk, sk, evk \leftarrow \mathrm{KeyGen}$

$c \leftarrow \mathrm{Enc}_{pk}(x)$

$Q(x) \leftarrow \mathrm{Dec}_{sk}(c')$

$\longleftarrow c'$

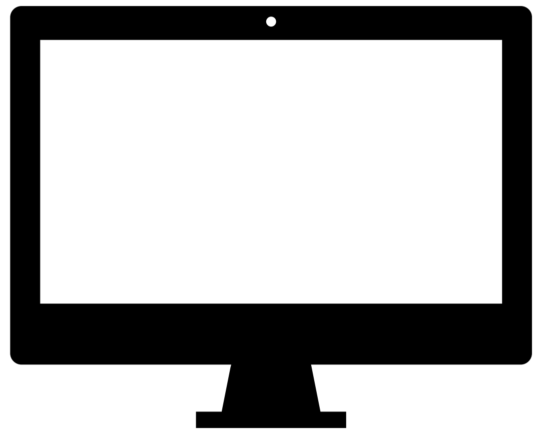
$c' \leftarrow \mathrm{Eval}(Q, c, evk)$

Client efficiency:

- Decryption should be more efficient than computing Q

Non-compact QFHE

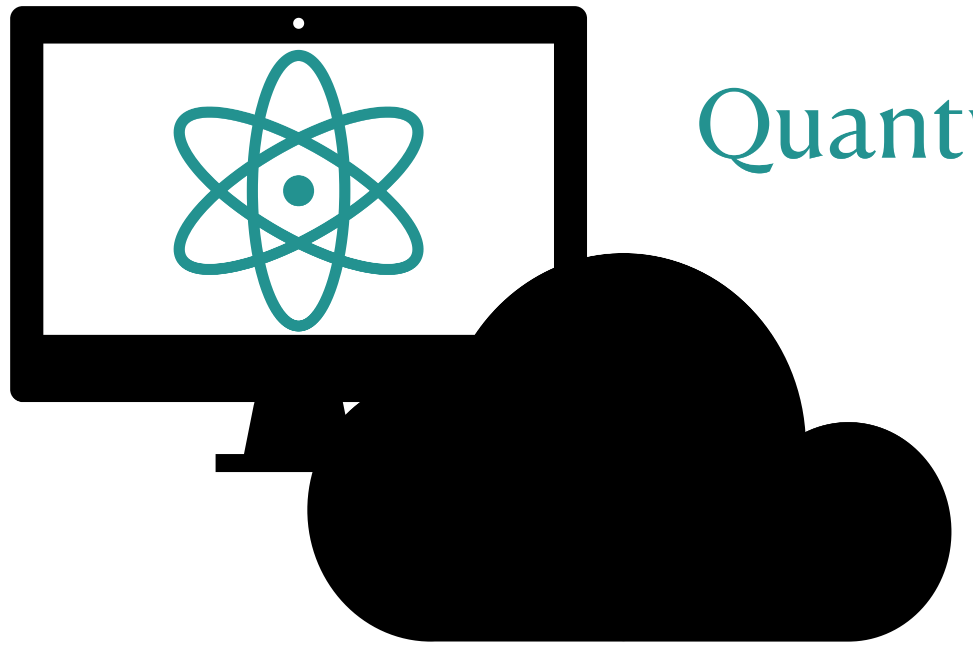
Input x



c, evk $\longrightarrow$

$\longrightarrow$ $\longleftarrow$ $\longrightarrow$ … $\longrightarrow$

c' $\longleftarrow$



Quantum circuit Q

$pk, sk, evk \leftarrow \text{KeyGen}$

$c \leftarrow \text{Enc}_{pk}(x)$

$Q(x) \leftarrow \text{Dec}_{sk}(c')$

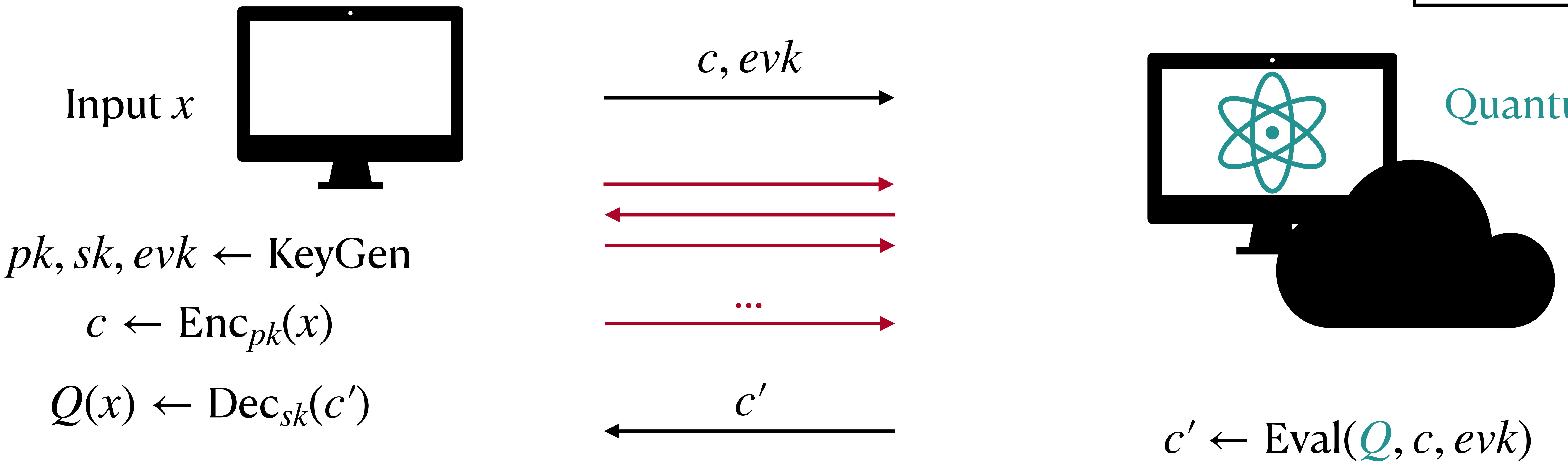
$c' \leftarrow \text{Eval}(Q, c, evk)$

Client efficiency:

- ~~Decryption should be more efficient than computing Q~~
  - Still useful for outsourcing quantum computation if we allow **interaction** and a **non-compact** poly($|Q|$)-time client
- Client should be classical

Non-compact QFHE

- Called Blind Delegated Quantum Computation
- Previously known only with LWE



Client efficiency:

- ~~Decryption should be more efficient than computing Q~~
  - Still useful for outsourcing quantum computation if we allow **interaction** and a **non-compact** poly($|Q|$)-time client
- Client should be classical

Our Results (Non-compact QFHE)

Theorem 1 [GV24] ++

pq. Dual-mode Trapdoor Functions $\Longrightarrow$ Non-compact QFHE

Our Results (Non-compact QFHE)

Theorem 1 [GV24] ++

pq. Dual-mode Trapdoor Functions $\Longrightarrow$ Non-compact QFHE

$\Uparrow$

Group actions **[GV24, Theorem 2]**

Our Results (Non-compact QFHE)

Theorem 1 [GV24] ++

pq. Dual-mode Trapdoor Functions $\Longrightarrow$ Non-compact QFHE

$\Uparrow$

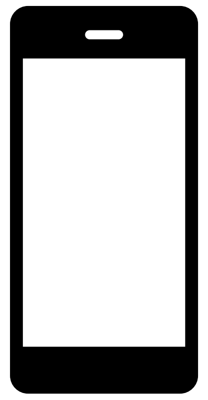
Group actions **[GV24, Theorem 2]**

Corollary [GV24]. Group actions $\Longrightarrow$ Non-compact QFHE

Not known to imply classical (compact) FHE

Our scheme for (compact) QFHE

Input x



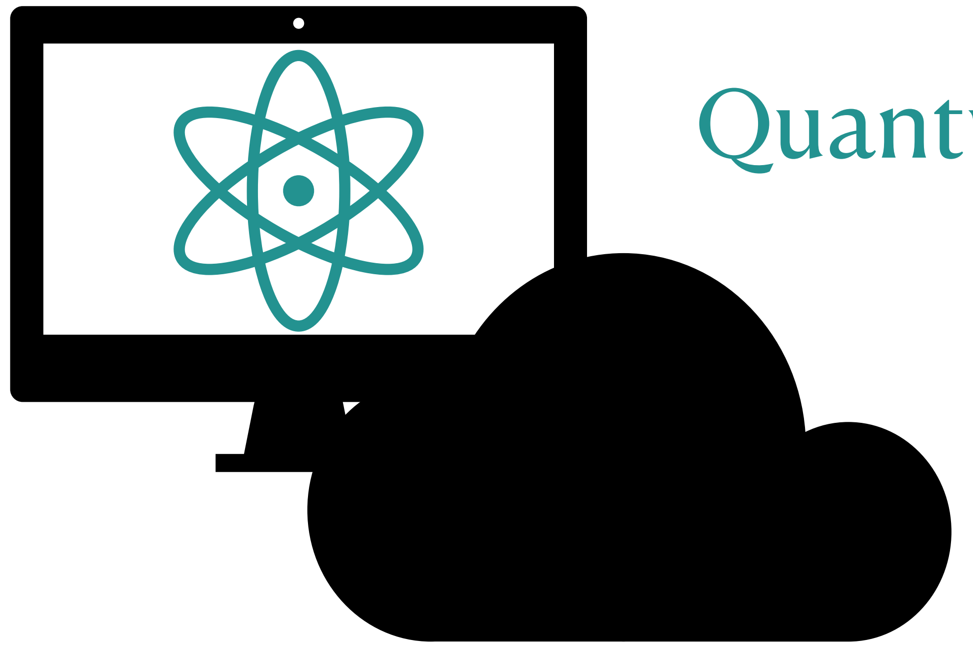
$pk, sk, evk \leftarrow \mathrm{KeyGen}$

$c \leftarrow \mathrm{Enc}_{pk}(x)$

$Q(x) \leftarrow \mathrm{Dec}_{sk}(c')$

$\xrightarrow{\quad c,\, evk \quad}$

$\xleftarrow{\quad c' \quad}$



Quantum circuit Q

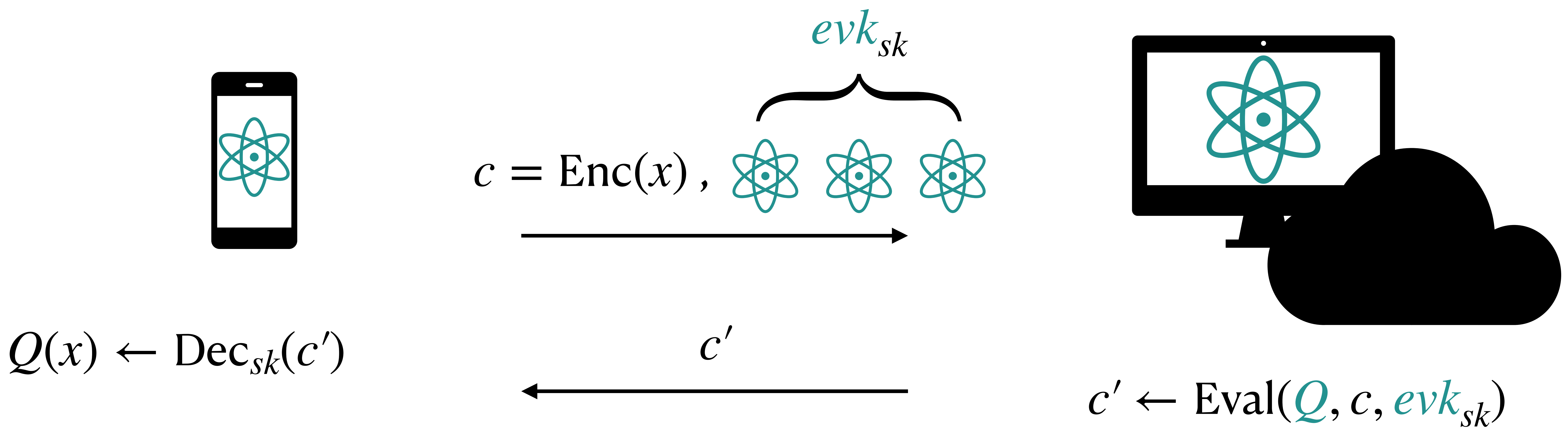
$c' \leftarrow \mathrm{Eval}\,(Q, c, evk)$

Server should not learn anything about x

Client efficiency:

- Decryption should be more efficient than computing Q
- **Client should be classical!**

Starting point: Dulek-Schaffner-Speelman'16

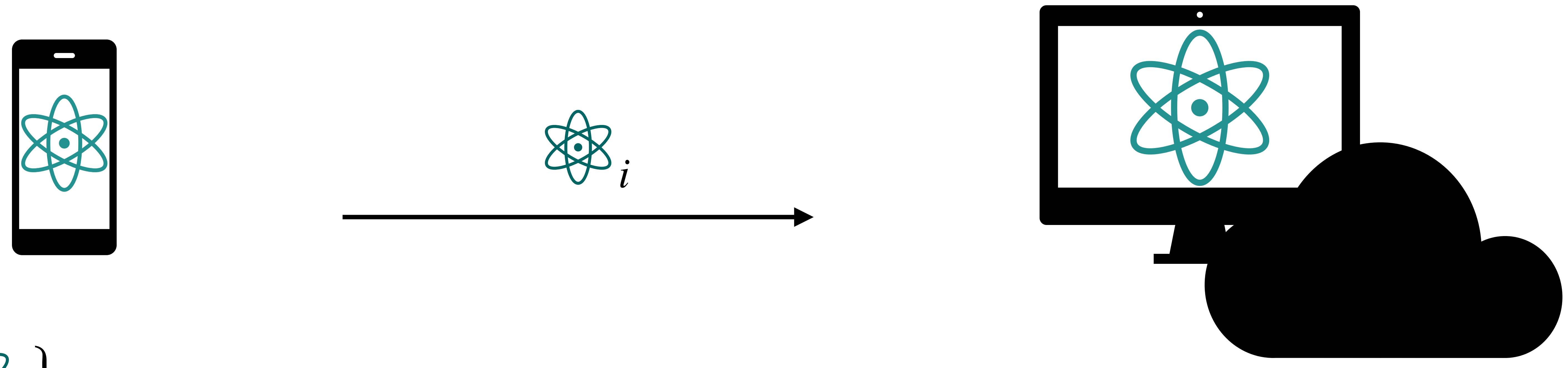


- Client needs to be quantum 🙁
- Needs to prepare and send quantum evaluation keys evk (which is a function of sk)

Our Idea: Use RSP

Remote State Preparation (RSP)

[Djunko-Kashefi'16, Gheorghiu-Vidick'19,
Cojocaru-Colisson-Kashefi-Wallden'19, Gheorghiu-Metger-Poremba'22]



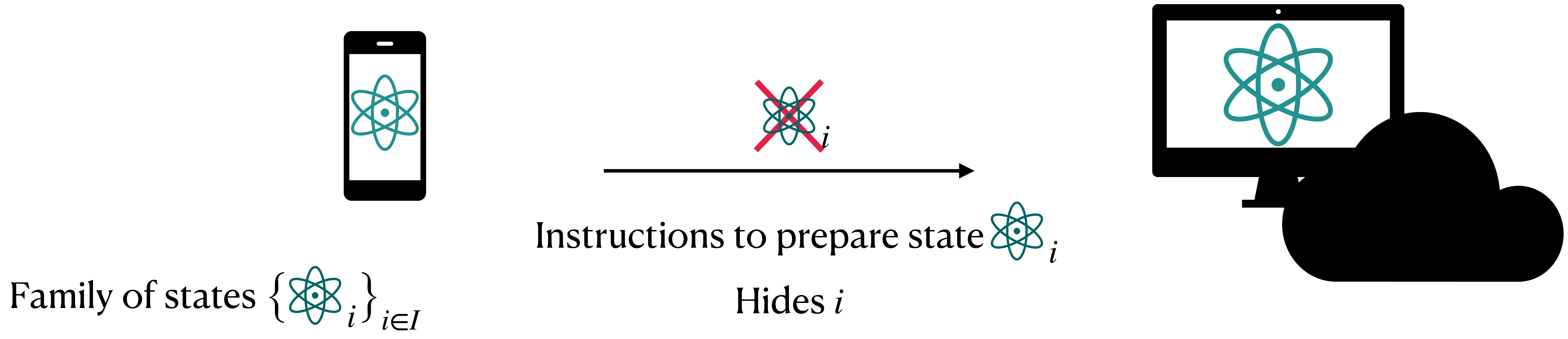
Family of states $\{\ {}_i\}_{i\in I}$

- Goal: Replace quantum communication in some protocols.

Our Idea: Use RSP

Remote State Preparation (RSP)

[Djunko-Kashefi'16, Gheorghiu-Vidick'19, Cojocaru-Colisson-Kashefi-Wallden'19, Gheorghiu-Metger-Poremba'22]

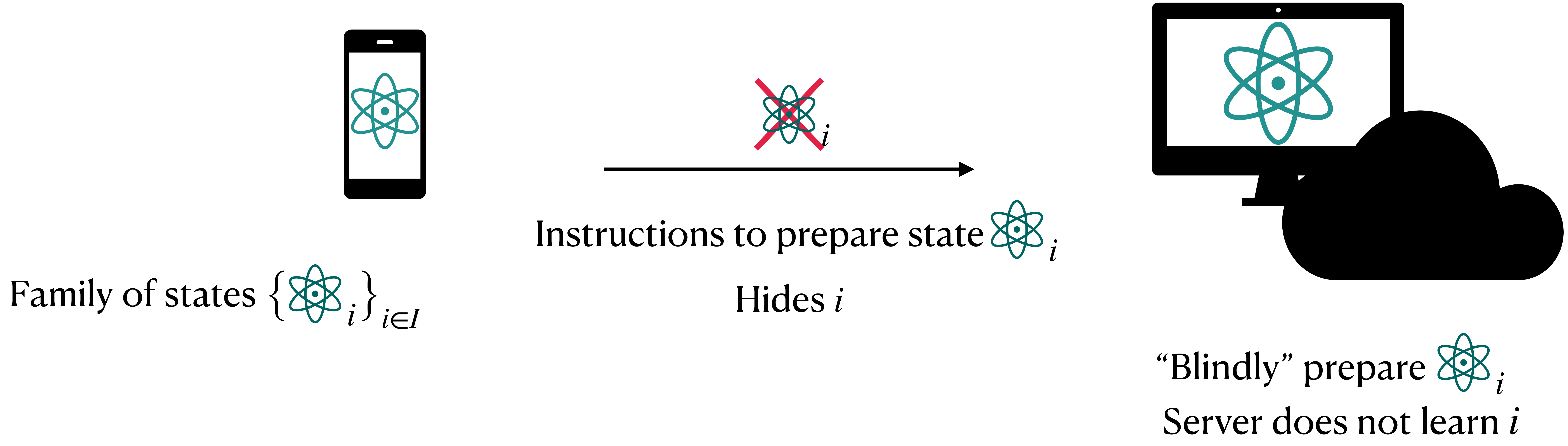


- Goal: Replace quantum communication in some protocols.

Our Idea: Use RSP

Remote State Preparation (RSP)

[Djunko-Kashefi'16, Gheorghiu-Vidick'19, Cojocaru-Colisson-Kashefi-Wallden'19, Gheorghiu-Metger-Poremba'22]

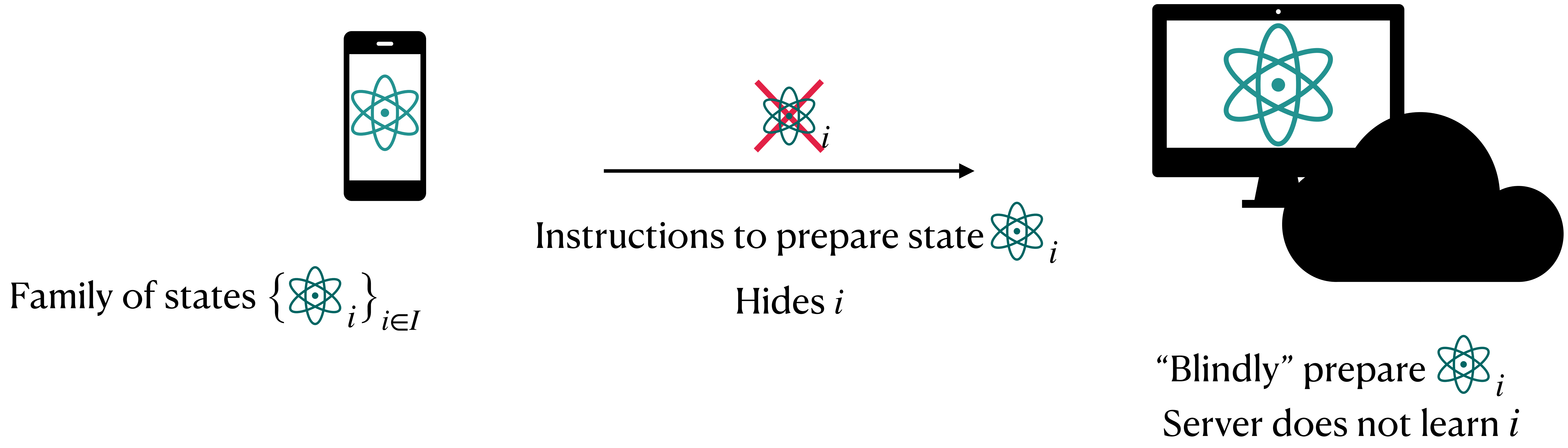


- Goal: Replace quantum communication in some protocols.

Our Idea: Use RSP

Remote State Preparation (RSP)

[Djunko-Kashefi'16, Gheorghiu-Vidick'19,
Cojocaru-Colisson-Kashefi-Wallden'19, Gheorghiu-Metger-Poremba'22]



- Goal: Replace quantum communication in some protocols.
- What is known? Can RSP BB84 states assuming dual-mode trapdoor functions [GV19, GMP22].

Our Main Technical Contribution

RSP the DSS evaluation keys

- Can we do remote state preparation for the DSS quantum *evk*?

Our Main Technical Contribution

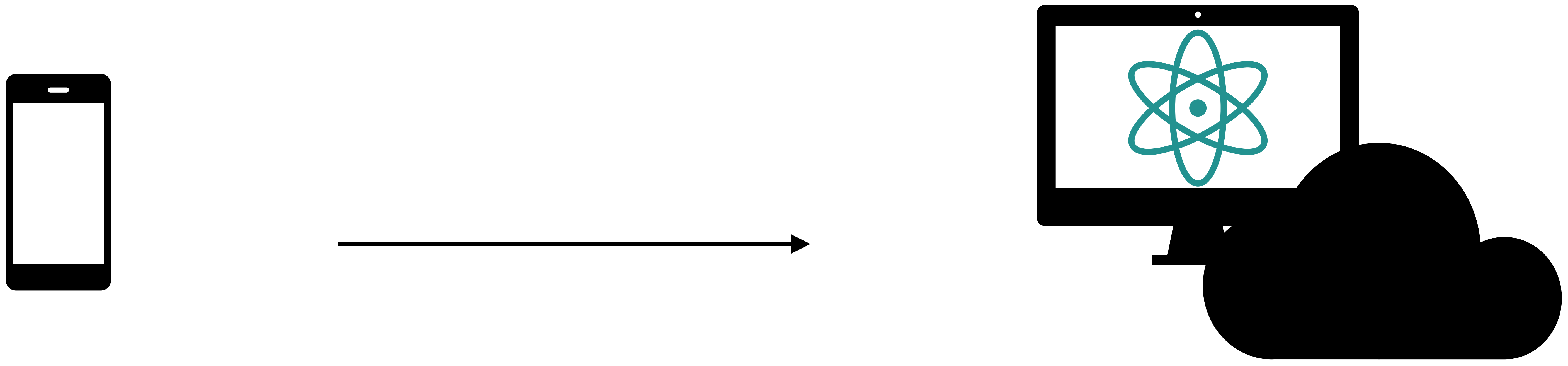
RSP the DSS evaluation keys

- Can we do remote state preparation for the DSS quantum *evk*?
- We show: Yes! Assuming post-quantum dual-mode trapdoor functions.

Our Main Technical Contribution

RSP the DSS evaluation keys

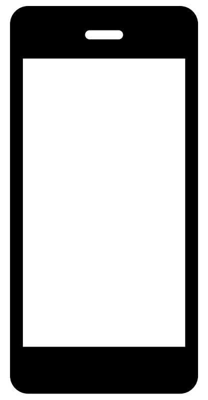
- Can we do remote state preparation for the DSS quantum *evk*?
- We show: Yes! Assuming post-quantum dual-mode trapdoor functions.



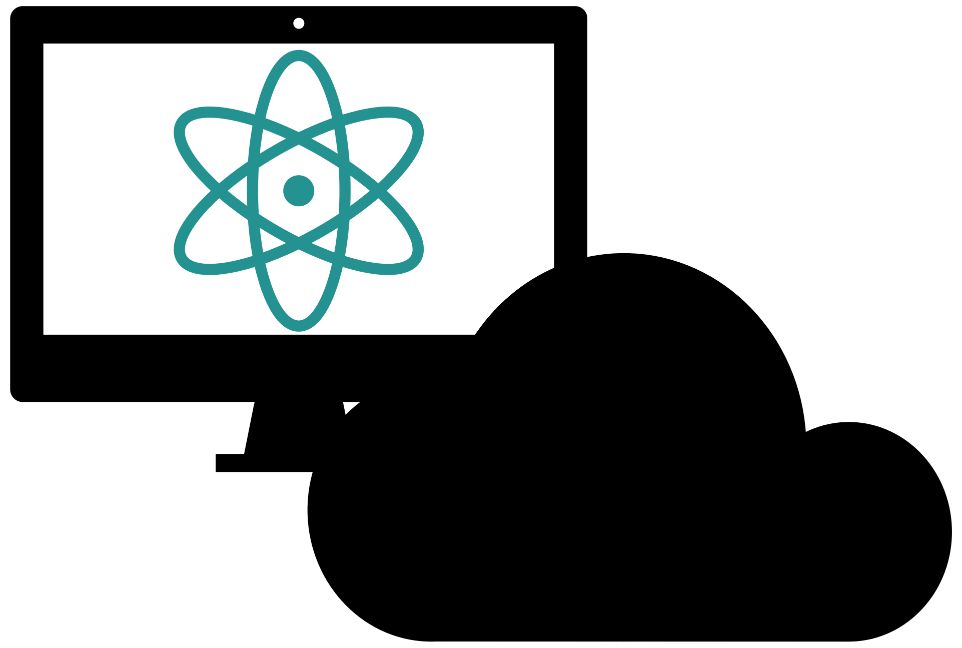
Our Main Technical Contribution

RSP the DSS evaluation keys

- Can we do remote state preparation for the DSS quantum *evk*?
- We show: Yes! Assuming post-quantum dual-mode trapdoor functions.



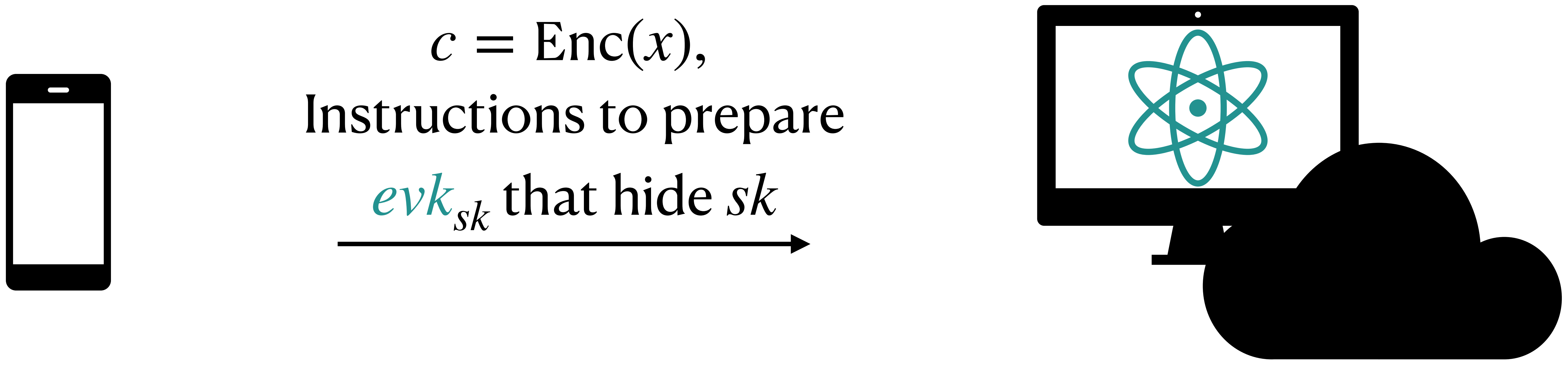
$c = \mathrm{Enc}(x)$,
Instructions to prepare
evk_{sk} that hide sk
$\longrightarrow$



Our Main Technical Contribution

RSP the DSS evaluation keys

- Can we do remote state preparation for the DSS quantum *evk*?
- We show: Yes! Assuming post-quantum dual-mode trapdoor functions.

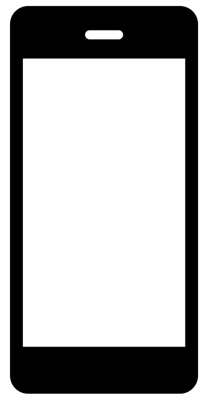


Prepares evk_{sk} but learns nothing about sk

Our Main Technical Contribution

RSP the DSS evaluation keys

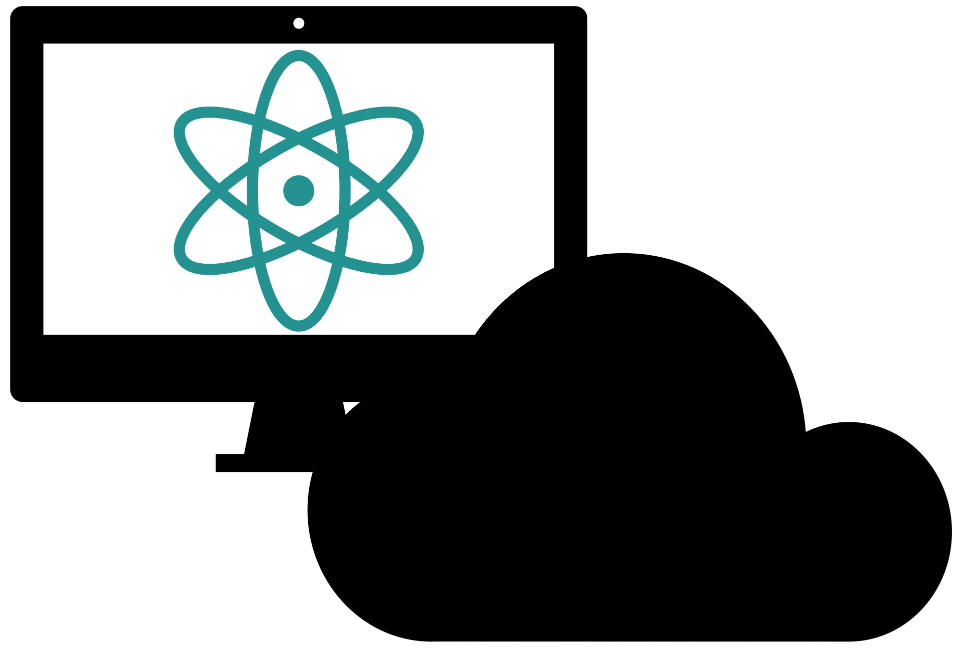
- Can we do remote state preparation for the DSS quantum *evk*?
- We show: Yes! Assuming post-quantum dual-mode trapdoor functions.



$c = \mathrm{Enc}(x)$,
Instructions to prepare
evk_{sk} that hide sk →

← c'

$Q(x) \leftarrow \mathrm{Dec}_{sk}(c')$



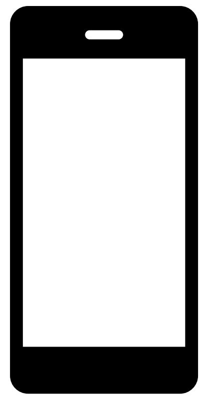
Prepares evk_{sk} but learns nothing about sk

$c' \leftarrow \mathrm{Eval}(Q, c, evk_{sk})$

Our Main Technical Contribution

RSP the DSS evaluation keys

- Can we do remote state preparation for the DSS quantum $\textcolor{teal}{evk}$?
- We show: Yes! Assuming post-quantum dual-mode trapdoor functions.

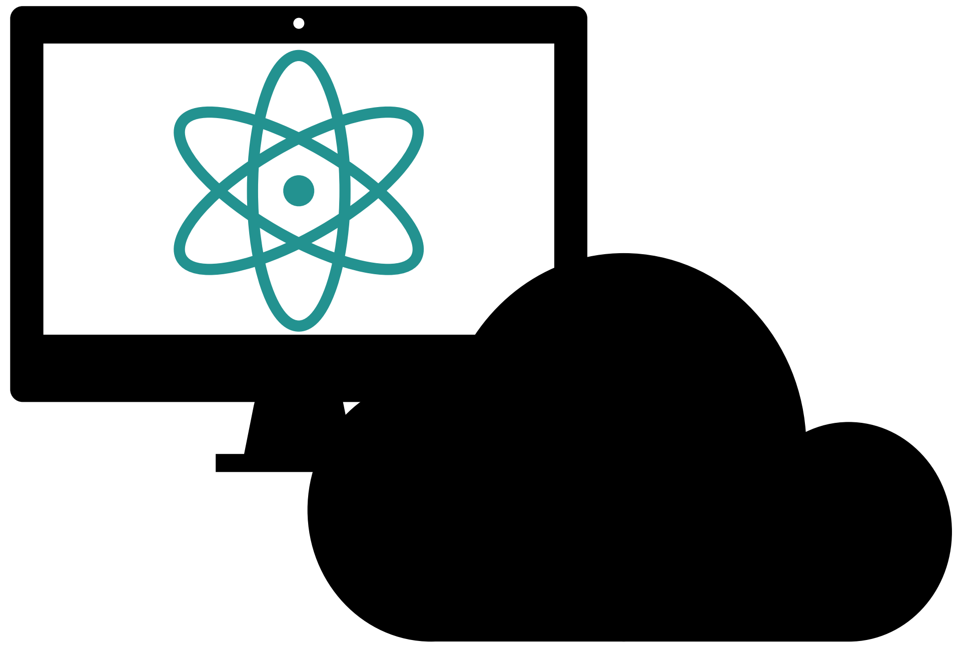


$c = \mathrm{Enc}(x)$,
Instructions to prepare
evk_{sk} that hide sk

$\longrightarrow$

c'

$\longleftarrow$



$Q(x) \leftarrow \mathrm{Dec}_{sk}(c')$

Prepares evk_{sk} but learns nothing about sk

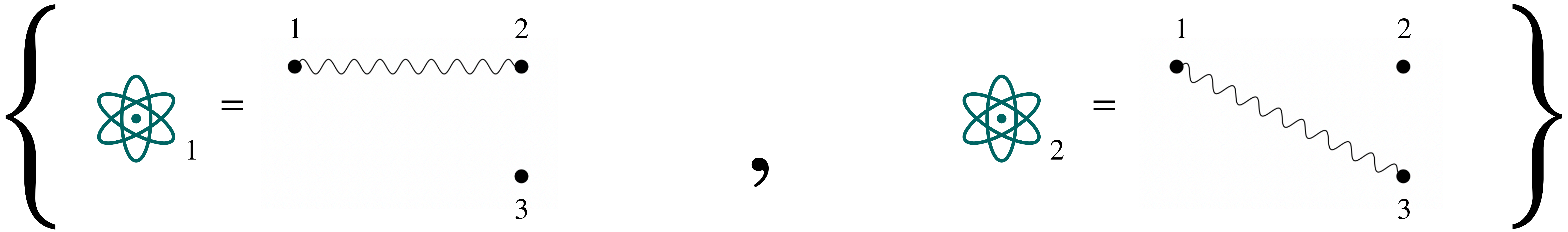
$c' \leftarrow \mathrm{Eval}(Q, c, evk_{sk})$

Client is classical!

Our Main Technical Contribution

RSP the DSS evaluation keys

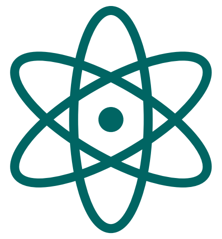
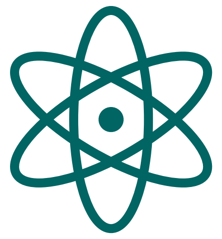
Step (1) It (essentially) suffices to RSP one of these two states:



Our Main Technical Contribution

RSP the DSS evaluation keys

Step (1) It (essentially) suffices to RSP one of these two states:

Step (2) Dual-mode trapdoor functions $\Longrightarrow$ RSP of  $_1$ or  $_2$.

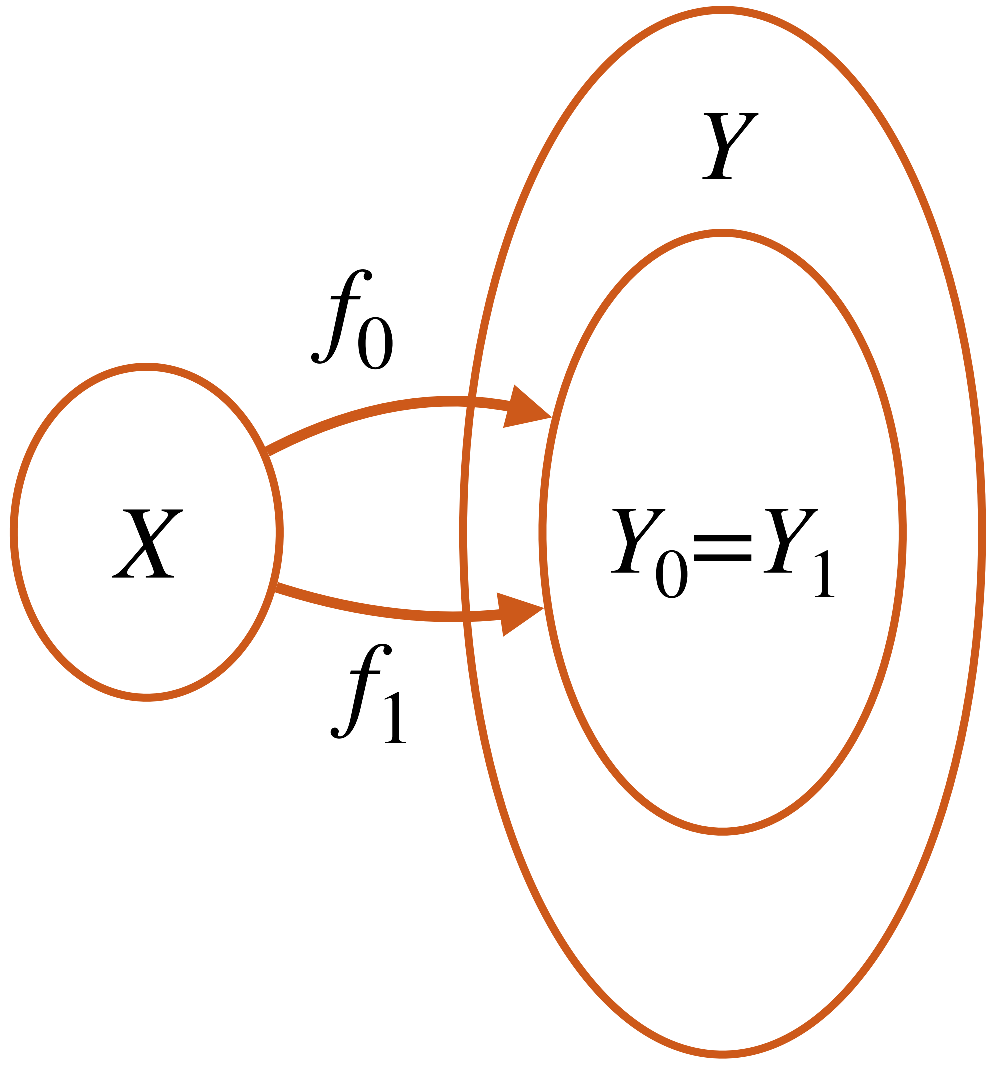
Dual-mode trapdoor functions

Definition. Family of pairs of injective functions $f_0, f_1 : X \to Y$ such that either

Dual-mode trapdoor functions

Definition. Family of pairs of injective functions $f_0, f_1 : X \to Y$ such that either

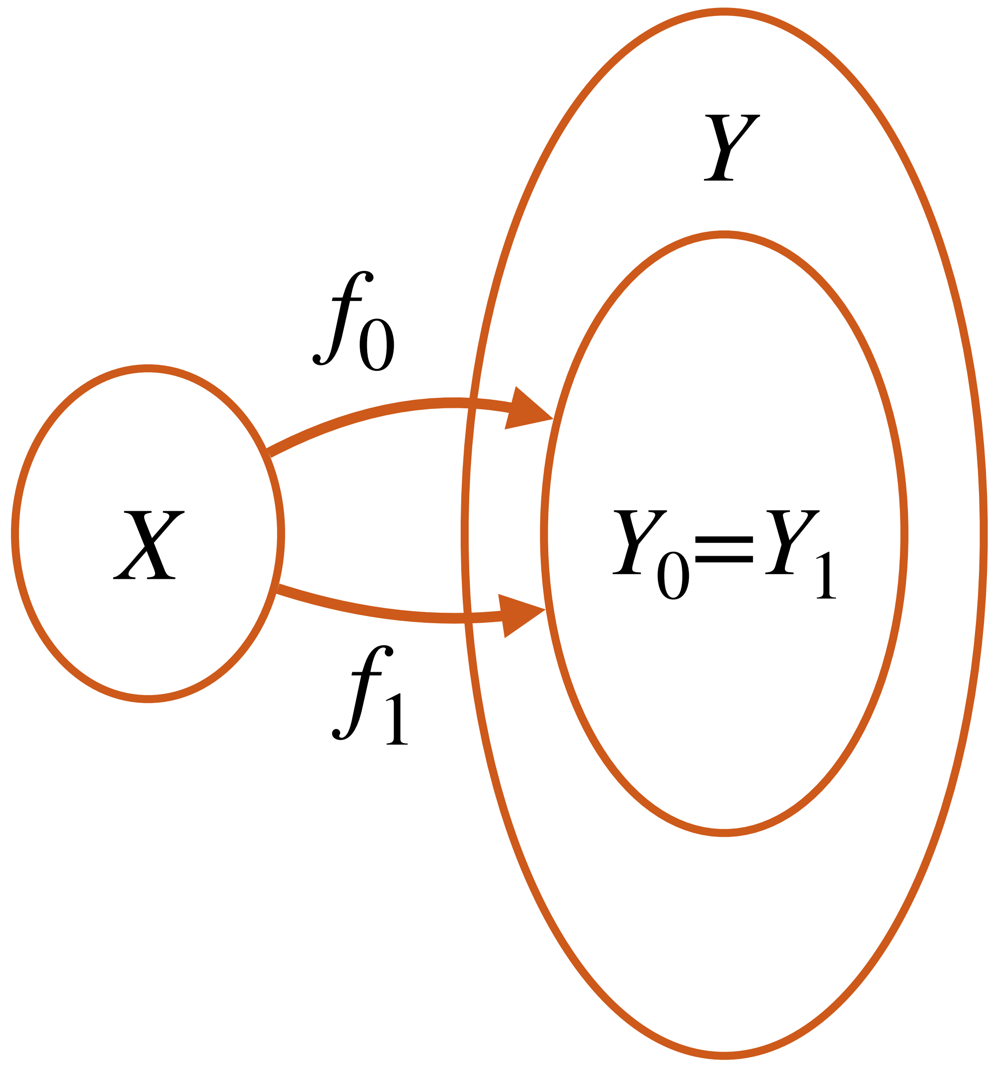
(1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$

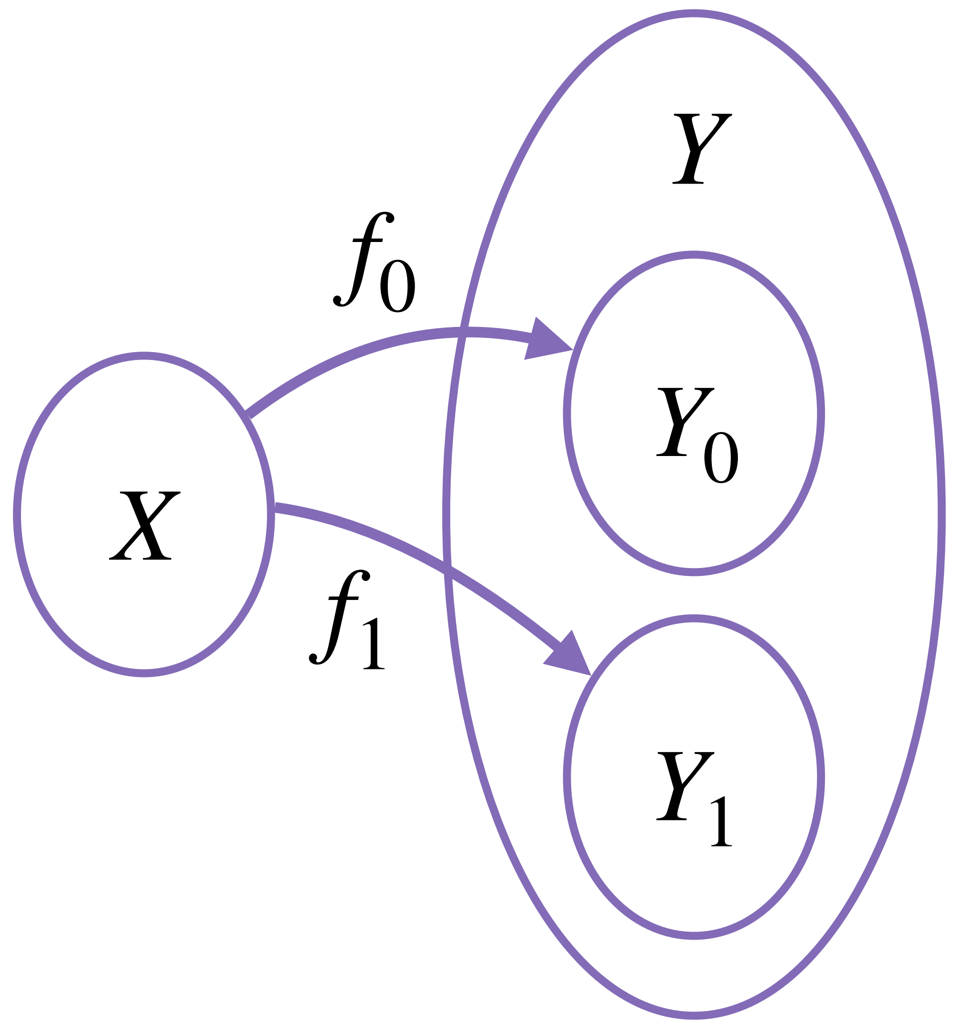


Dual-mode trapdoor functions

Definition. Family of pairs of injective functions $f_0, f_1 : X \to Y$ such that either

(1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$ or (2) $\mathrm{Im}(f_0) \cap \mathrm{Im}(f_1) = \varnothing$.

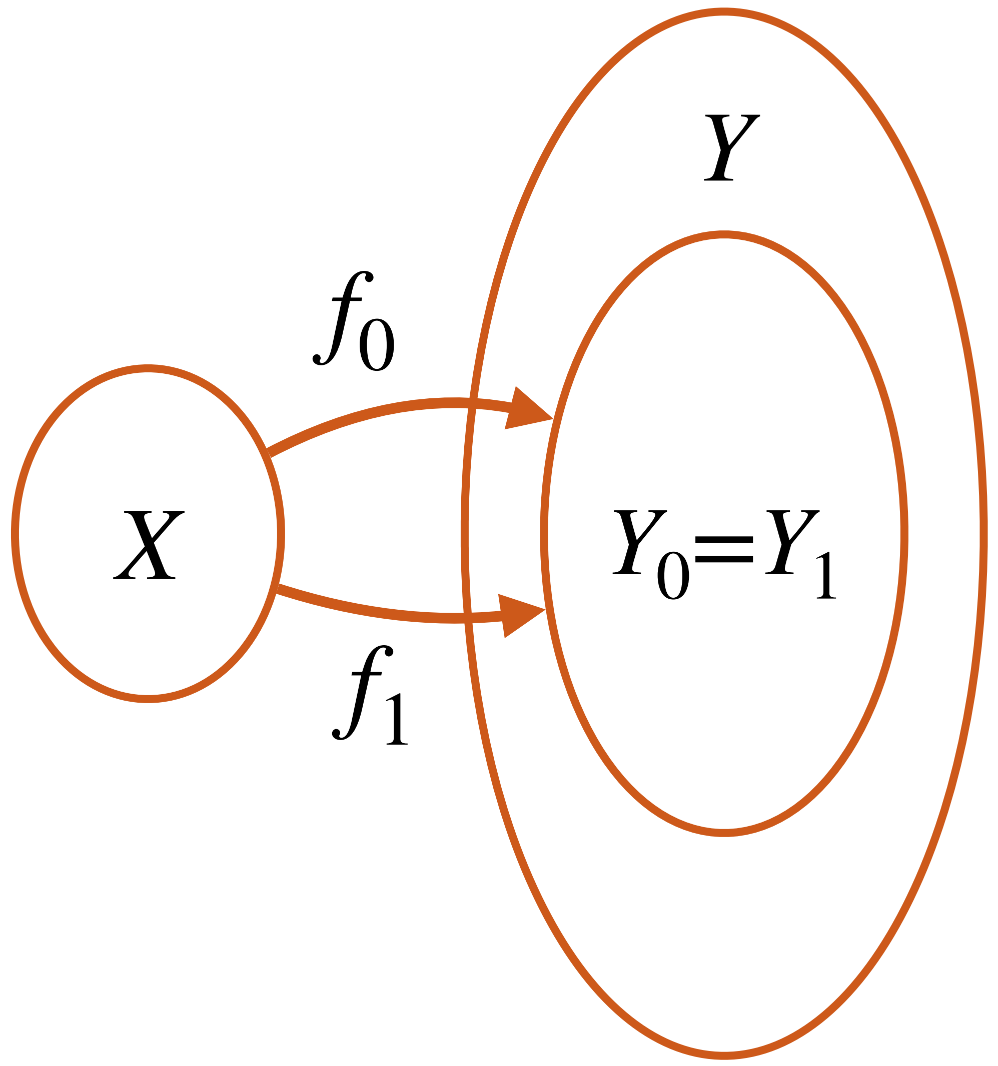




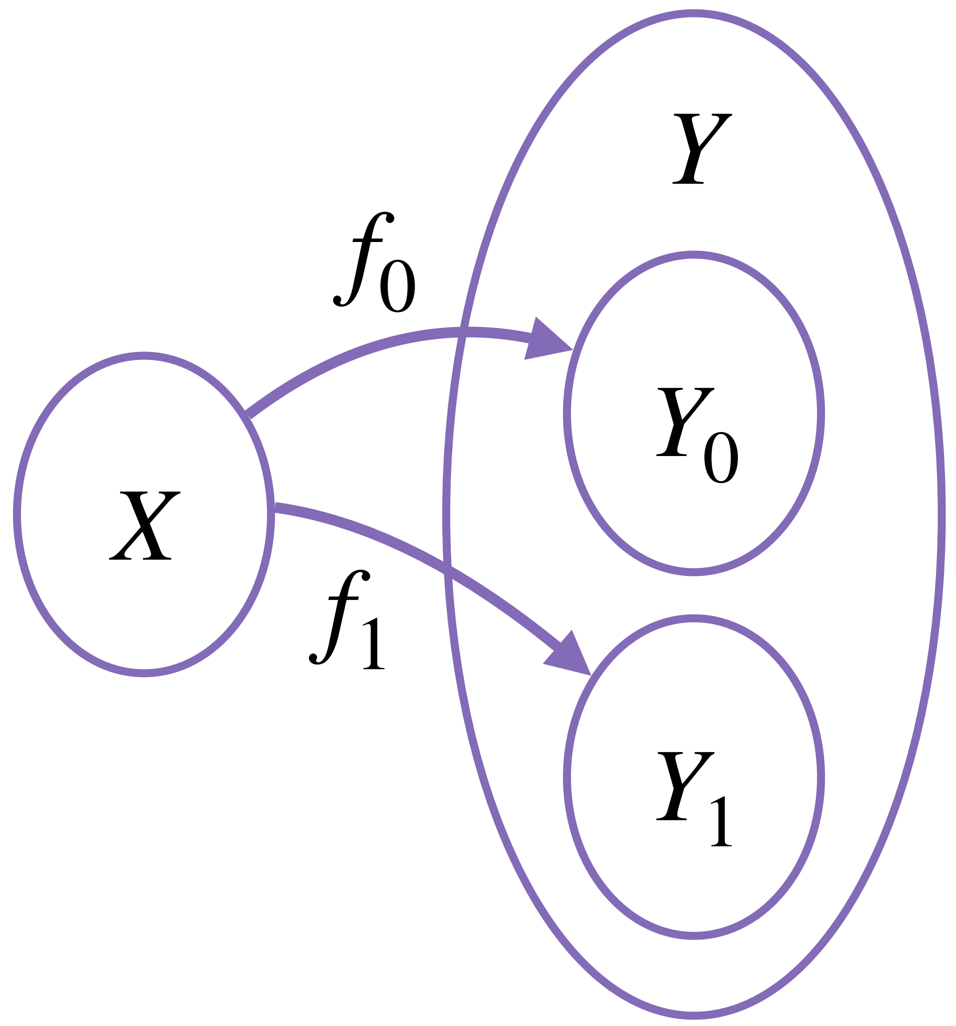
Dual-mode trapdoor functions

Definition. Family of pairs of injective functions $f_0, f_1 : X \to Y$ such that either

(1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$ or (2) $\mathrm{Im}(f_0) \cap \mathrm{Im}(f_1) = \varnothing$.



$\approx_c$

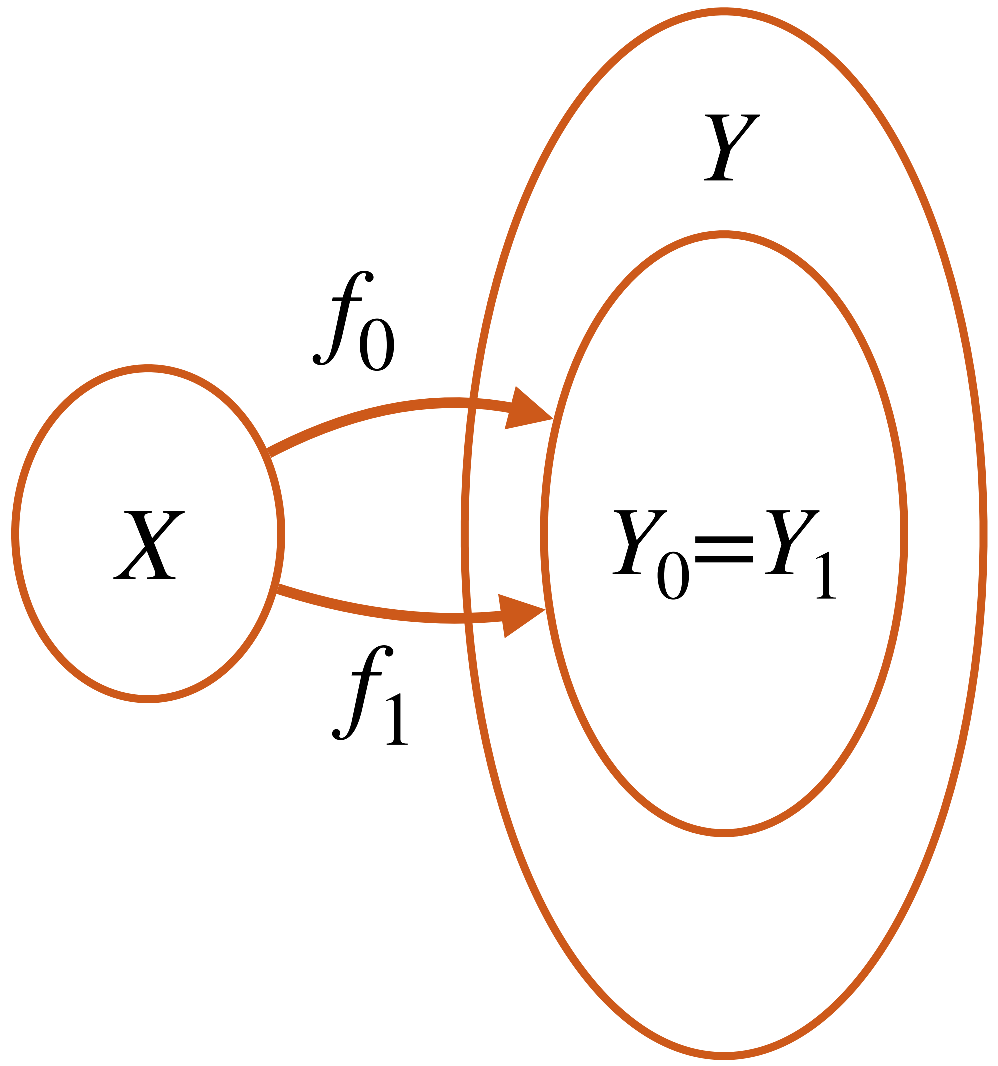


- **Dual-mode:** Given keys for f_0, f_1, it is hard to tell whether they are in mode (1) or (2).

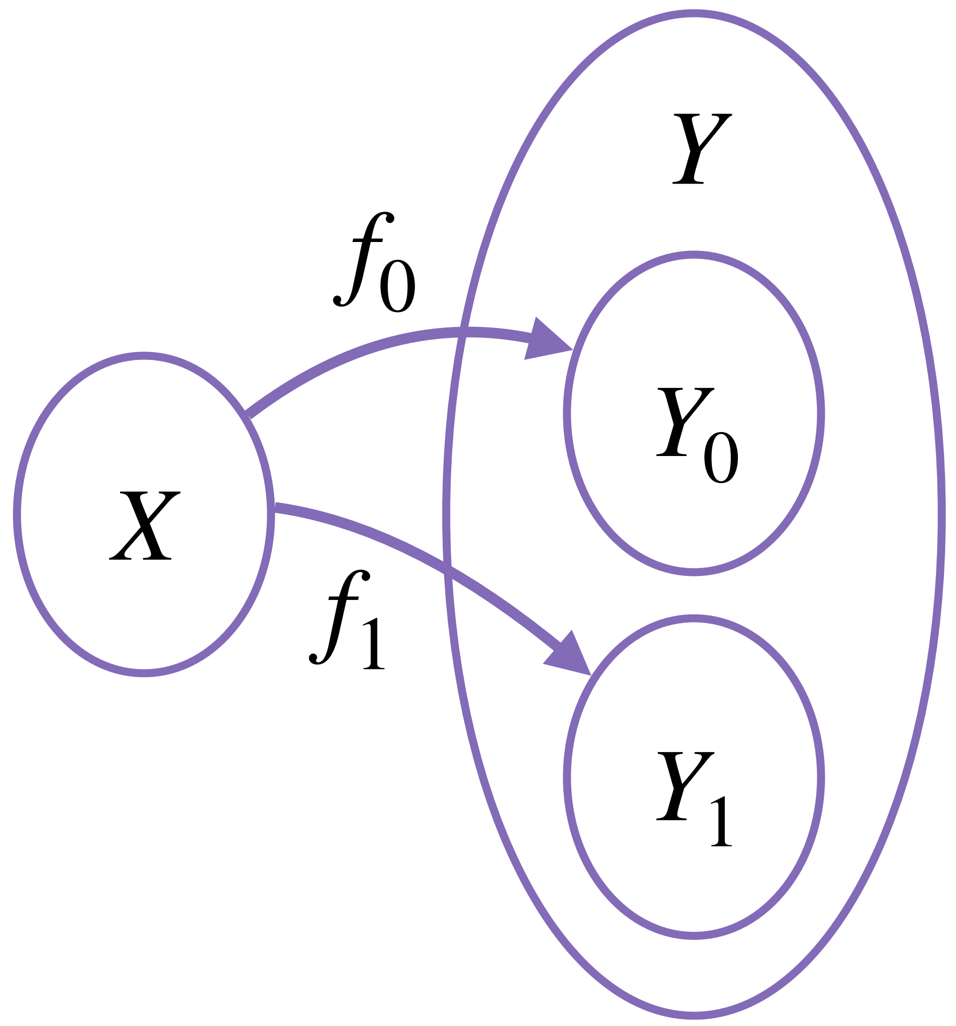
Dual-mode trapdoor functions

Definition. Family of pairs of injective functions $f_0, f_1 : X \to Y$ such that either

(1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$ or (2) $\mathrm{Im}(f_0) \cap \mathrm{Im}(f_1) = \varnothing$.



$\approx_c$



- **Dual-mode:** Given keys for f_0, f_1, it is hard to tell whether they are in mode (1) or (2).
- **Trapdoor:** allows efficient inversion in both modes.

Warmup: Remote State Preparation of BB84 states

[GV19, GMP22]

Warmup: Remote State Preparation of BB84 states

[GV19, GMP22]

BB84 states: $\left\{ |0\rangle, \quad |1\rangle, \quad \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \quad \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \right\}$

$_1 \qquad _2 \qquad _3 \qquad _4$

Warmup: Remote State Preparation of BB84 states

[GV19, GMP22]

BB84 states: $\left\{ |0\rangle, \quad |1\rangle, \quad \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \quad \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \right\}$

1. Prepare state $\sum_{b\in\{0,1\}} \sum_{x\in X} |b\rangle |x\rangle$

Warmup: Remote State Preparation of BB84 states

[GV19, GMP22]

BB84 states: $\left\{ |0\rangle, \quad |1\rangle, \quad \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \quad \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \right\}$

1. Prepare state $\sum_{b\in\{0,1\}} \sum_{x\in X} |b\rangle\, |x\rangle\, |f_b(x)\rangle$

Warmup: Remote State Preparation of BB84 states

[GV19, GMP22]

BB84 states: $\left\{ |0\rangle, \quad |1\rangle, \quad \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big), \quad \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \right\}$

$_1 \qquad _2 \qquad _3 \qquad _4$

1. Prepare state $\sum_{b\in\{0,1\}} \sum_{x\in X} |b\rangle\,|x\rangle\,|f_b(x)\rangle$

2. Measure register #3 to get $y \in Y$

- In mode (1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$

$$|0\rangle\,|x_0\rangle + |1\rangle\,|x_1\rangle$$

- In mode (2) $\mathrm{Im}(f_0) \cap \mathrm{Im}(f_1) = \varnothing$

$$|b\rangle\,|x_b\rangle$$

Warmup: Remote State Preparation of BB84 states

[GV19, GMP22]

BB84 states: $\left\{ \underset{{}_1}{|0\rangle}, \quad \underset{{}_2}{|1\rangle}, \quad \frac{1}{\sqrt{2}}\underset{{}_3}{(|0\rangle + |1\rangle)}, \quad \frac{1}{\sqrt{2}}\underset{{}_4}{(|0\rangle - |1\rangle)} \right\}$

1. Prepare state $\sum_{b\in\{0,1\}} \sum_{x\in X} |b\rangle |x\rangle |f_b(x)\rangle$

2. Measure register #3 to get $y \in Y$

   - In mode (1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$

     $|0\rangle|x_0\rangle + |1\rangle|x_1\rangle$

   - In mode (2) $\mathrm{Im}(f_0) \cap \mathrm{Im}(f_1) = \varnothing$

     $|b\rangle|x_b\rangle$

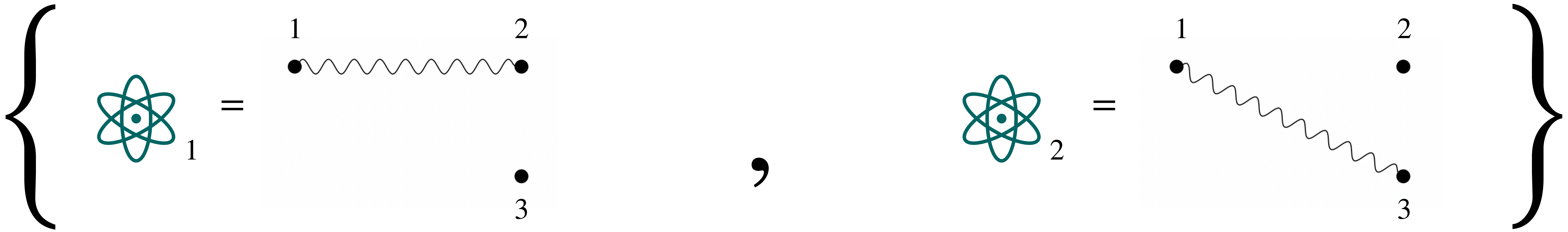
3. Measure register #2 in Hadamard basis to get rid of x_b

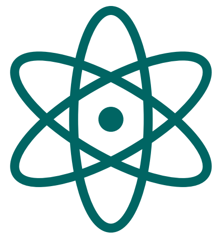
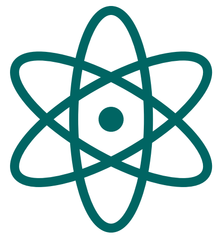
   - In mode (1) $\mathrm{Im}(f_0) = \mathrm{Im}(f_1)$ $\quad {}_3 \quad {}_4$
   - In mode (2) $\mathrm{Im}(f_0) \cap \mathrm{Im}(f_1) = \varnothing$ $\quad {}_1 \quad {}_2$

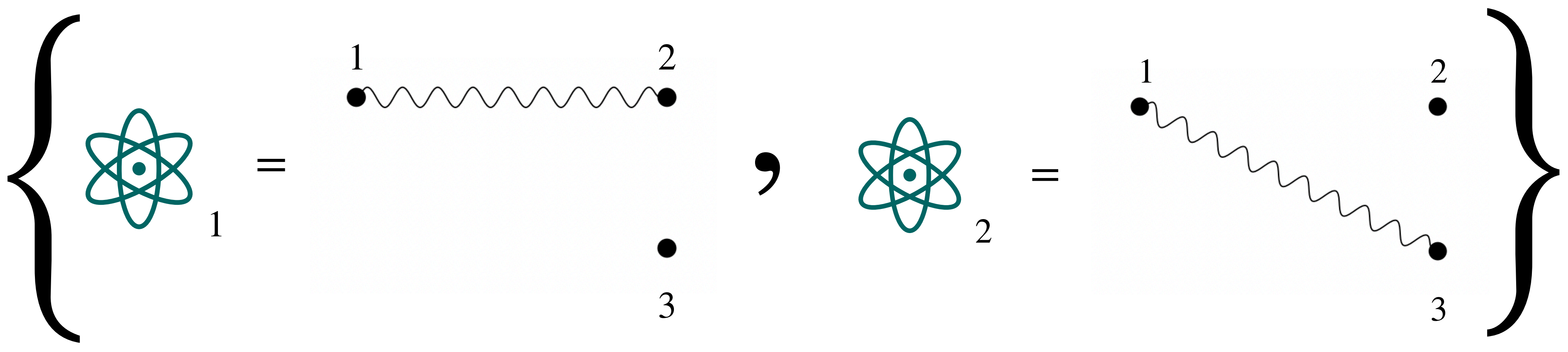
Our Main Technical Contribution

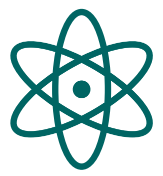
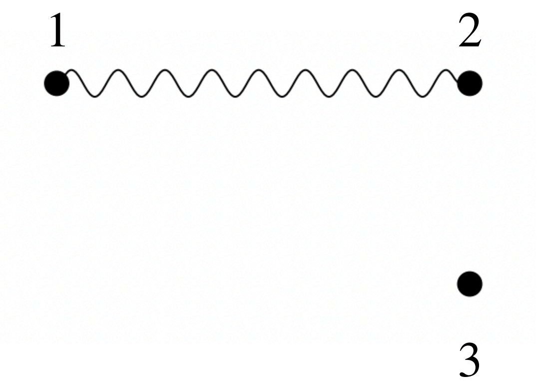
RSP the DSS evaluation keys

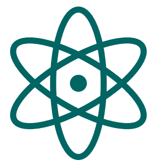
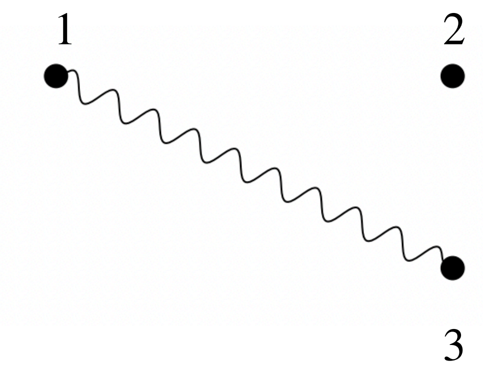
Step (1) It (essentially) suffices to RSP one of these two states:



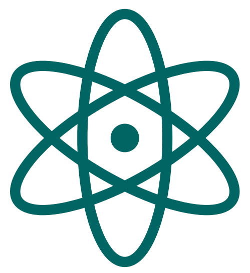
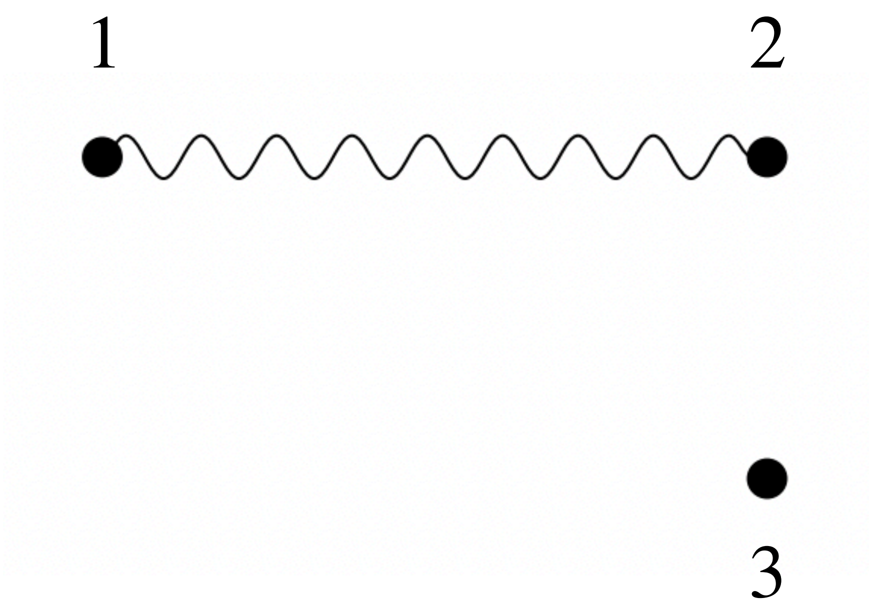
Step (2) Dual-mode trapdoor functions $\Longrightarrow$ RSP of  $_1$ or  $_2$.

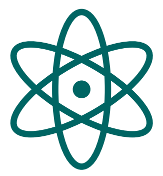
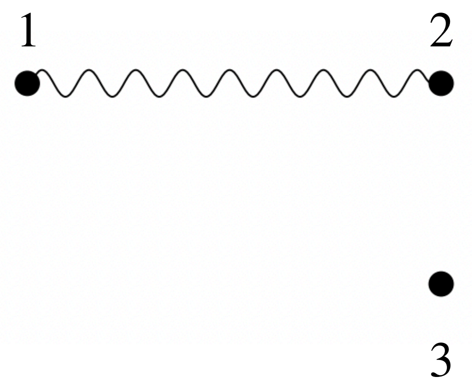
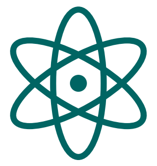
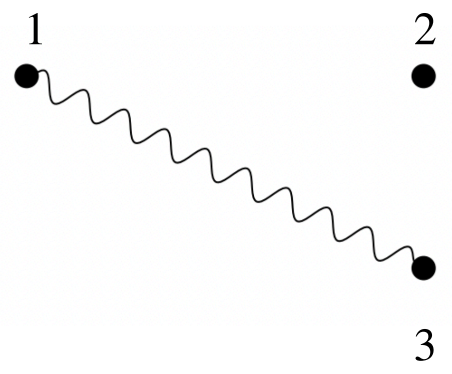
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for 

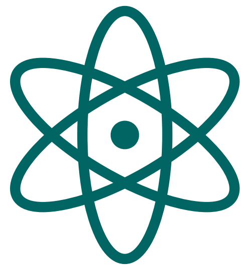
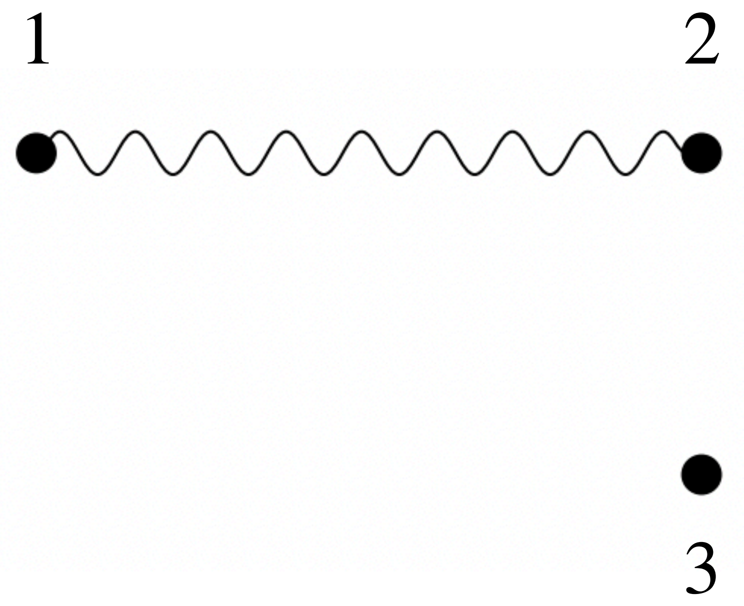
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\Big\{$  $_1$ = 

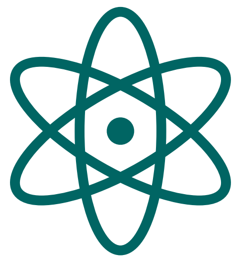
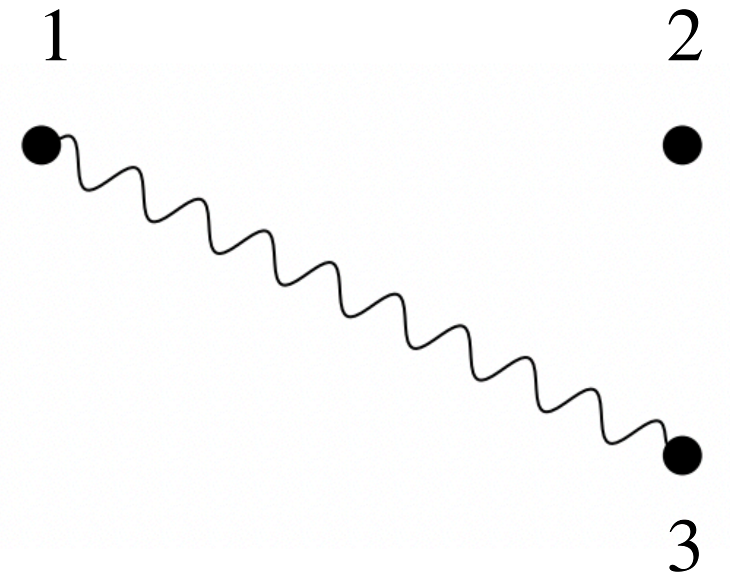
,  $_2$ = 

$\Big\}$

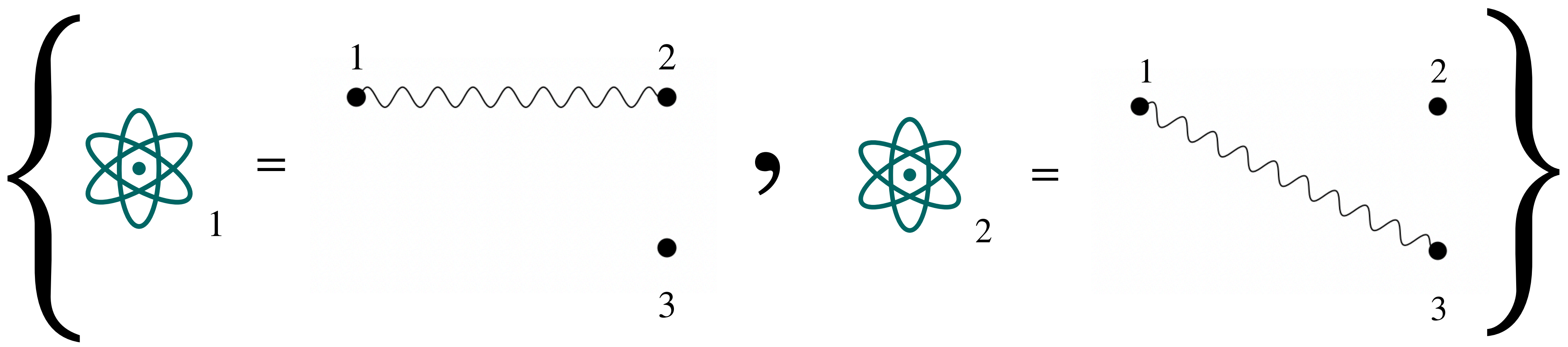
 $_1 = \frac{1}{2}\Big(|000\rangle + |001\rangle + |110\rangle + |111\rangle \Big) = \sum_{u=v, w\in\{0,1\}} |uvw\rangle$ = 

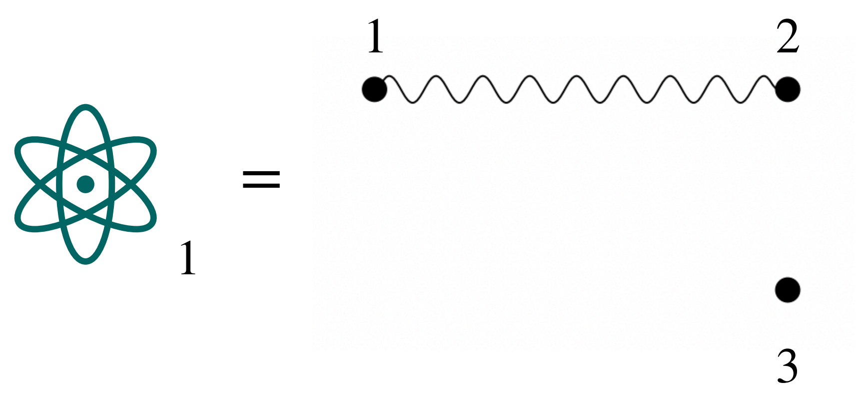
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\left\{ \right.$  $_1 =$  ,  $_2 =$  $\left. \right\}$

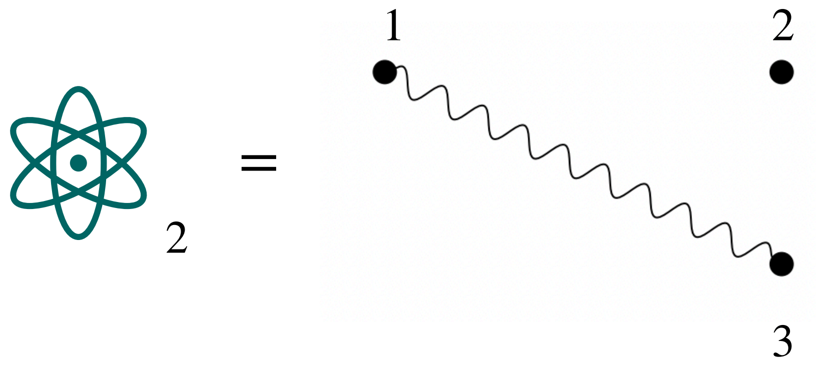
 $_1$ $= \frac{1}{2}\Big(|000\rangle + |001\rangle + |110\rangle + |111\rangle \Big) = \sum_{u=v, w\in\{0,1\}} |uvw\rangle$ $=$ 

 $_2$ $= \frac{1}{2}\Big(|000\rangle + |010\rangle + |101\rangle + |111\rangle \Big) = \sum_{u=w, v\in\{0,1\}} |uvw\rangle$ $=$ 

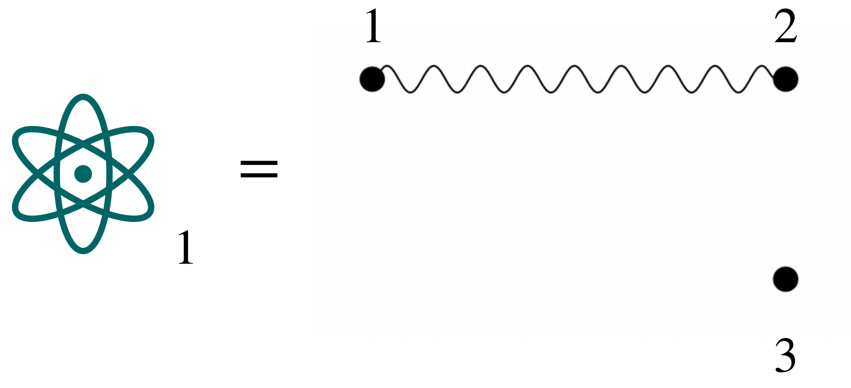
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for

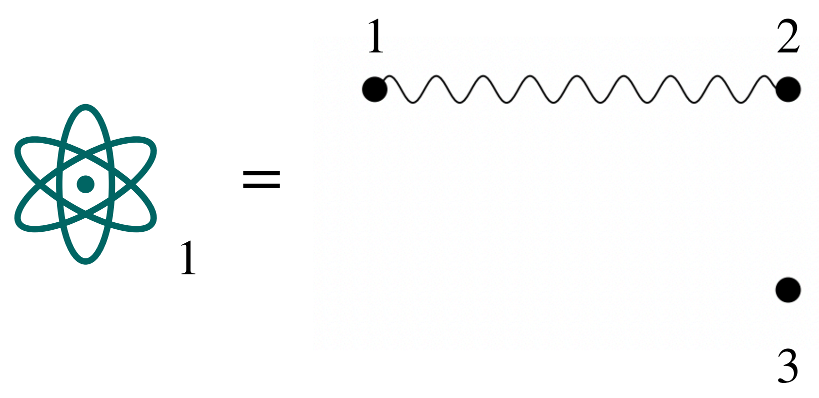


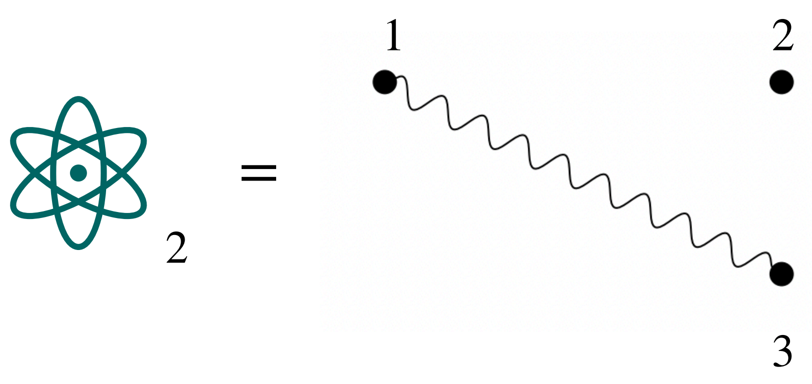
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\Big\{$ 

, 

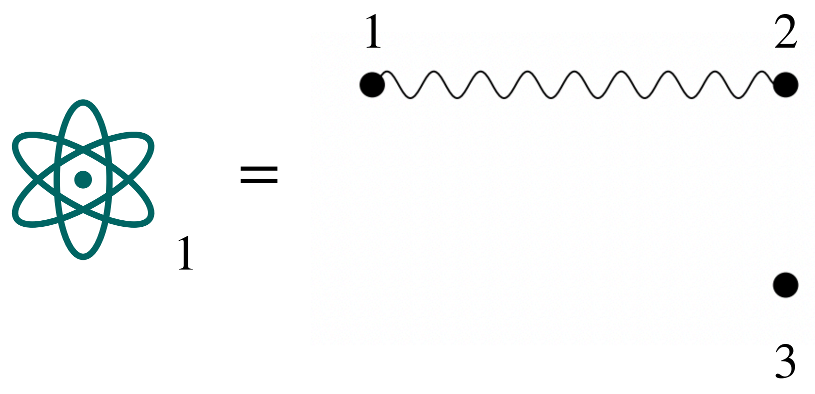
$\Big\}$

If client wants to prepare 

Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\Big\{$ 

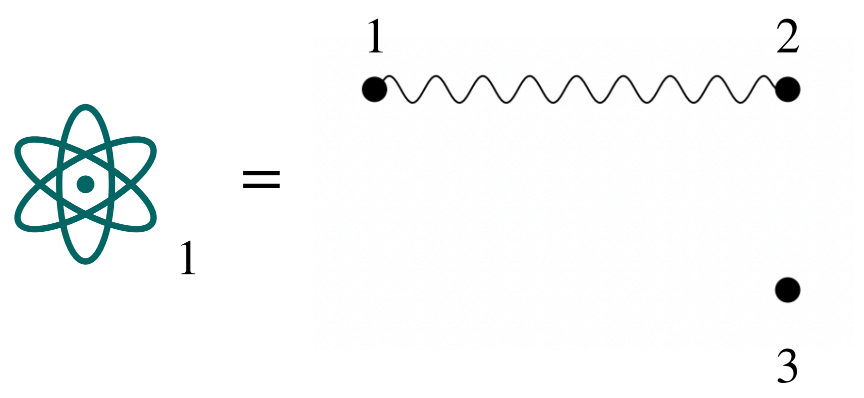
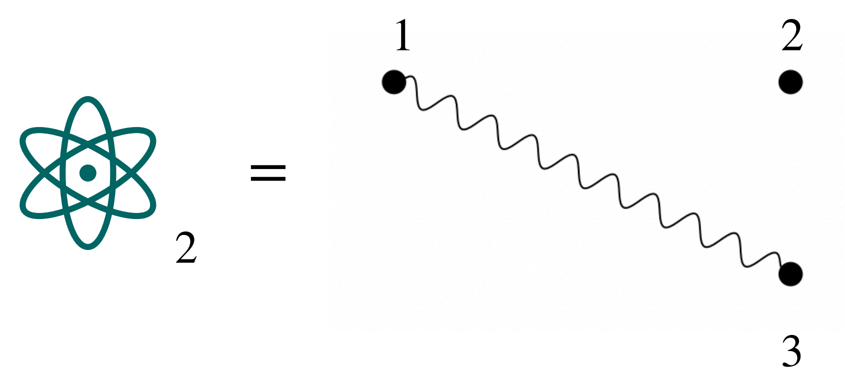
, 

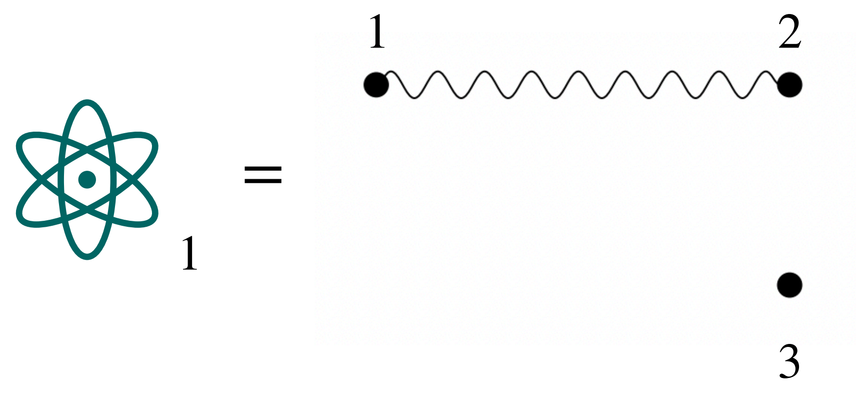
$\Big\}$

If client wants to prepare 

want to project to the blue basis vectors.

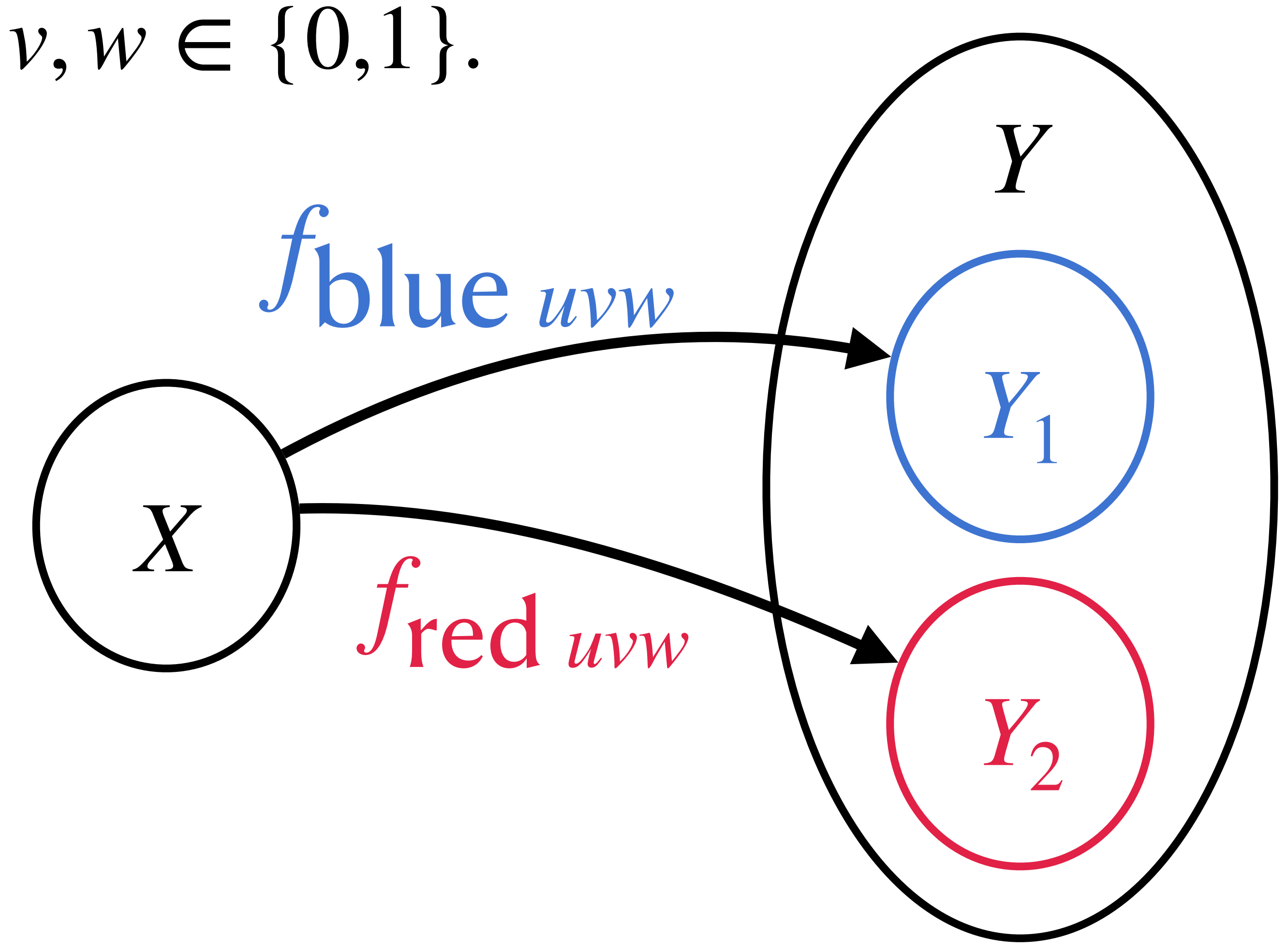
$$\frac{1}{2\sqrt{2}}\Big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle \Big)$$

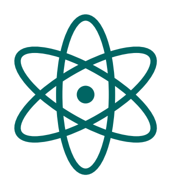
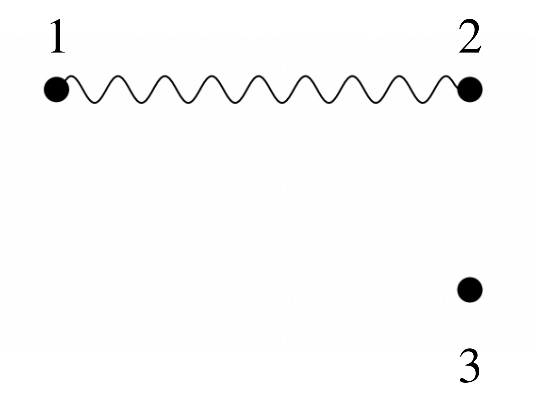
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\Big\{$  ,  $\Big\}$

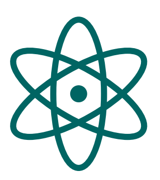
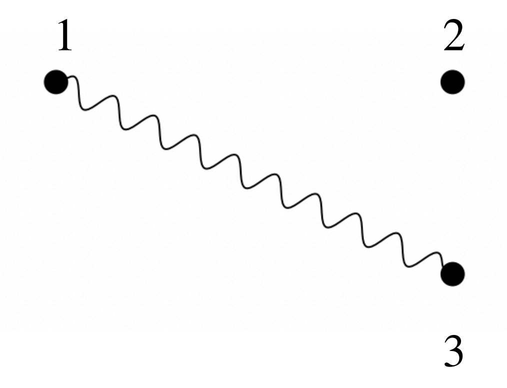
If client wants to prepare  want to project to the blue basis vectors.

$$\frac{1}{2\sqrt{2}}\Big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big)$$

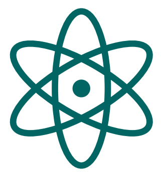
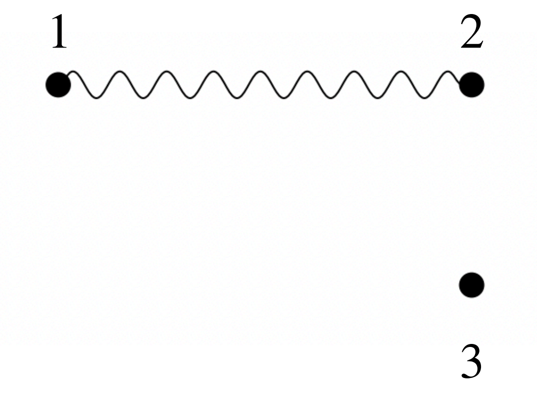
1. Prepare $\displaystyle\sum_{u,v,w\in\{0,1\}}\sum_{x\in X}|uvw\rangle|x\rangle|f_{uvw}(x)\rangle$ for 8 functions $f_{uvw} : X \to Y, u, v, w \in \{0,1\}$.



Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\Big\{$  $_1$ = 

,  $_2$ = 

$\Big\}$

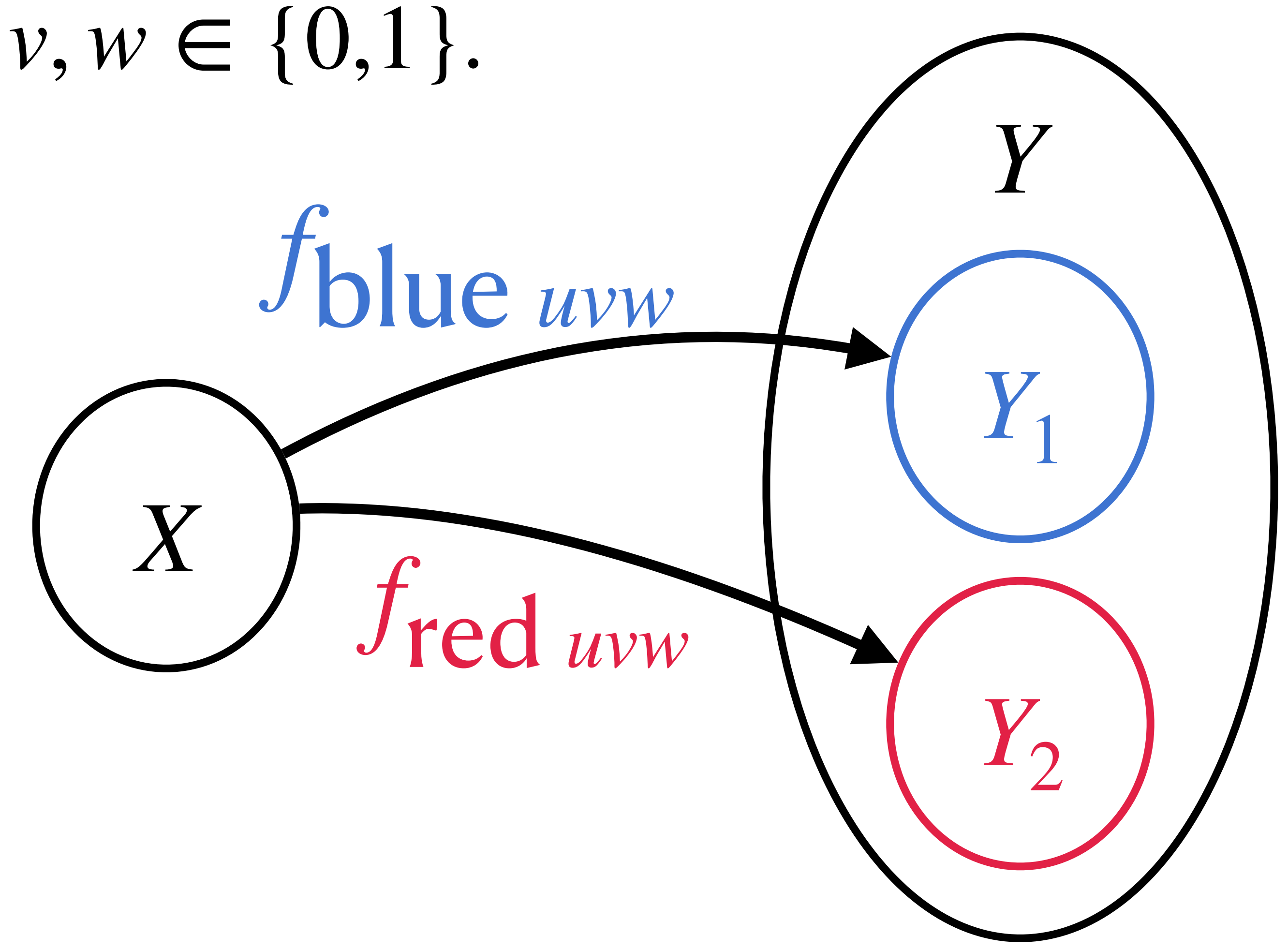
If client wants to prepare  $_1$ = 

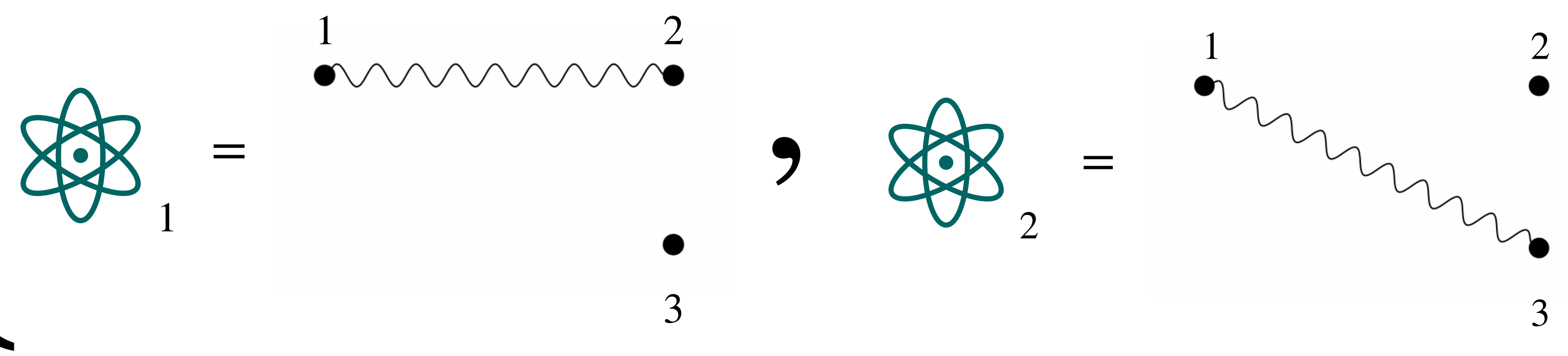
want to project to the blue basis vectors.

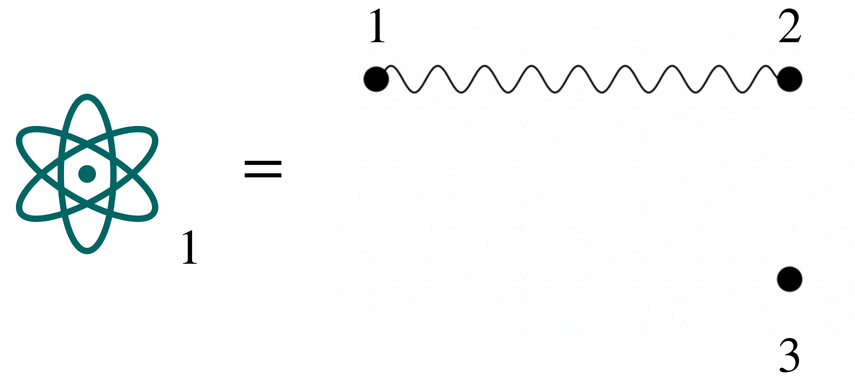
$$\frac{1}{2\sqrt{2}}\left(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\right)$$

1. Prepare $\sum_{u,v,w\in\{0,1\}} \sum_{x\in X} |uvw\rangle|x\rangle|f_{uvw}(x)\rangle$ for 8 functions $f_{uvw} : X \to Y, u, v, w \in \{0,1\}$.

2. Measure 3rd register, get $y \in Y_1$, $\sum_{\text{blue } u,v,w} |uvw\rangle|x_{uvw}\rangle$.



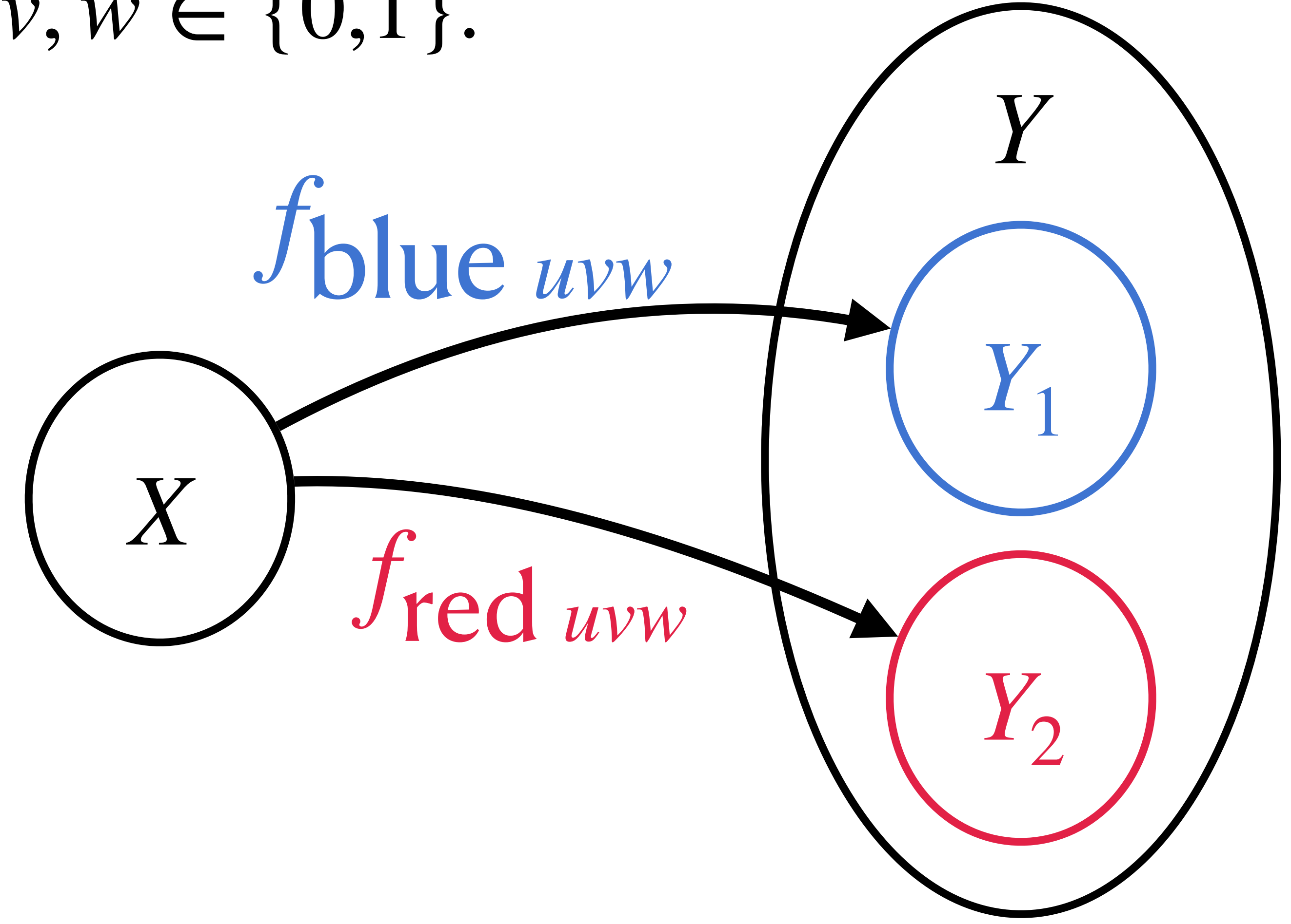
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\{$  $\}$

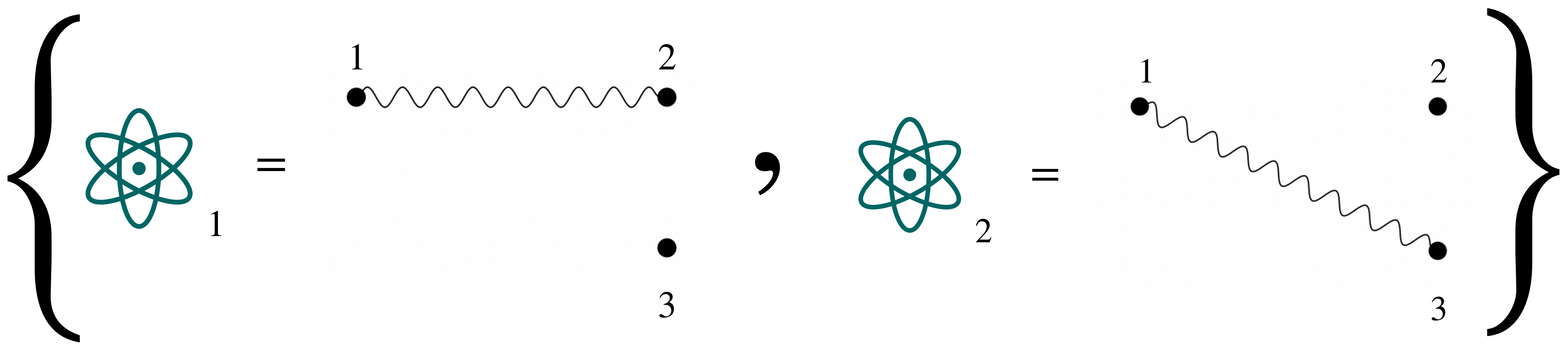
If client wants to prepare  want to project to the blue basis vectors.

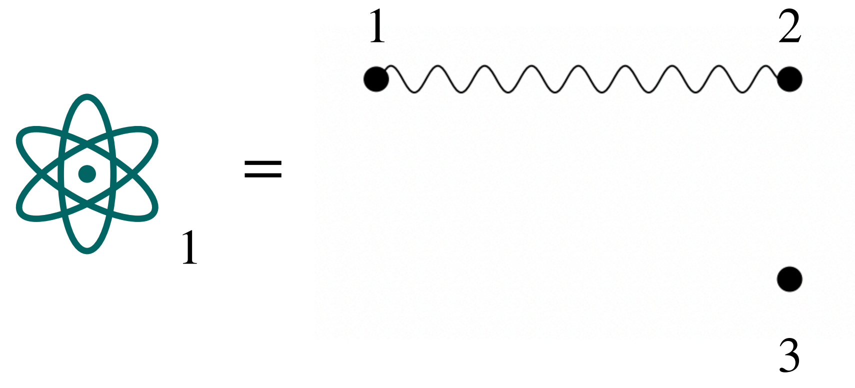
$$\frac{1}{2\sqrt{2}}\Big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big)$$

1. Prepare $\displaystyle\sum_{u,v,w\in\{0,1\}}\sum_{x\in X}|uvw\rangle|x\rangle|f_{uvw}(x)\rangle$ for 8 functions $f_{uvw}: X \to Y, u, v, w \in \{0,1\}$.

2. Measure 3rd register, get $y \in Y_1$, $\sum_{\text{blue } u,v,w}|uvw\rangle|x_{uvw}\rangle$.



Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for 

If client wants to prepare 

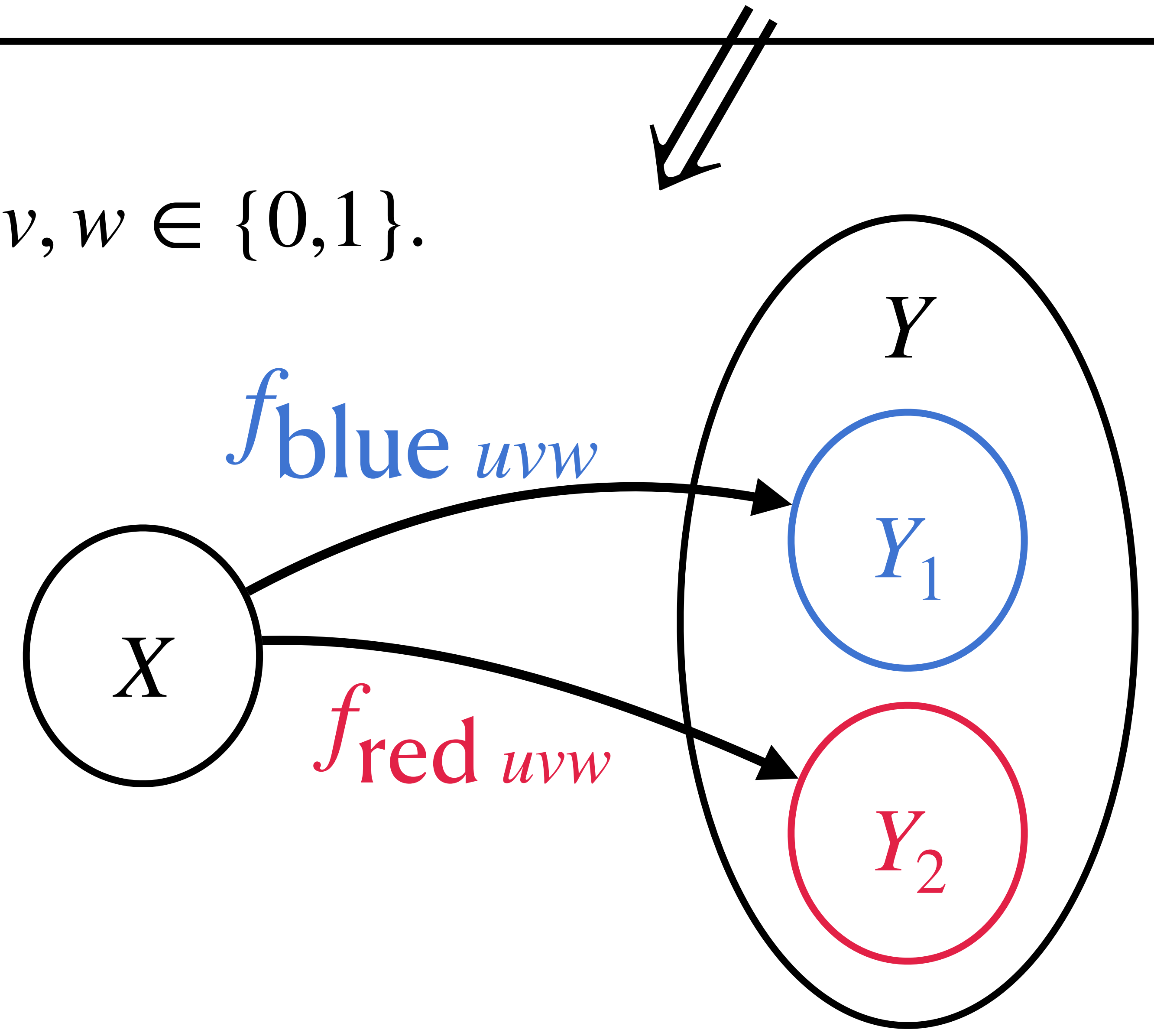
want to project to the blue basis vectors.

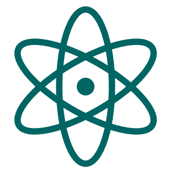
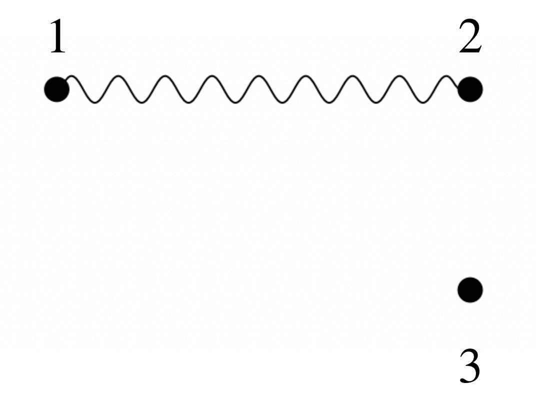
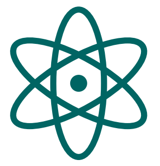
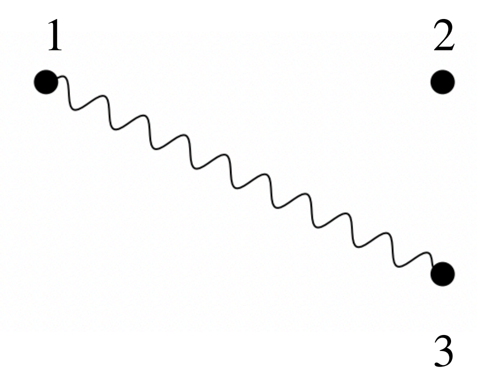
$$\frac{1}{2\sqrt{2}}\Big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big)$$

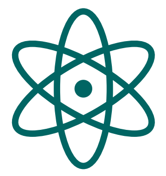
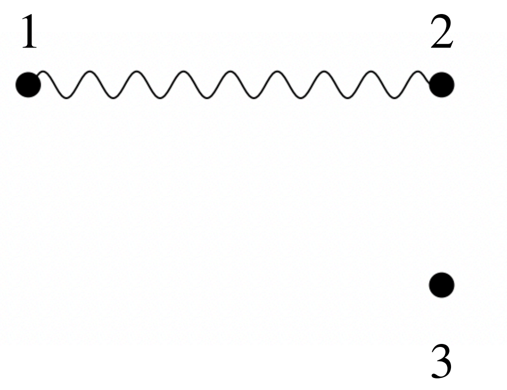
Dual-mode trapdoor functions

1. Prepare $\sum_{u,v,w\in\{0,1\}}\sum_{x\in X}|uvw\rangle|x\rangle|f_{uvw}(x)\rangle$ for 8 functions $f_{uvw}: X \to Y, u, v, w \in \{0,1\}$.

2. Measure 3rd register, get $y \in Y_1$, $\sum_{\text{blue } u,v,w}|uvw\rangle|x_{uvw}\rangle$.



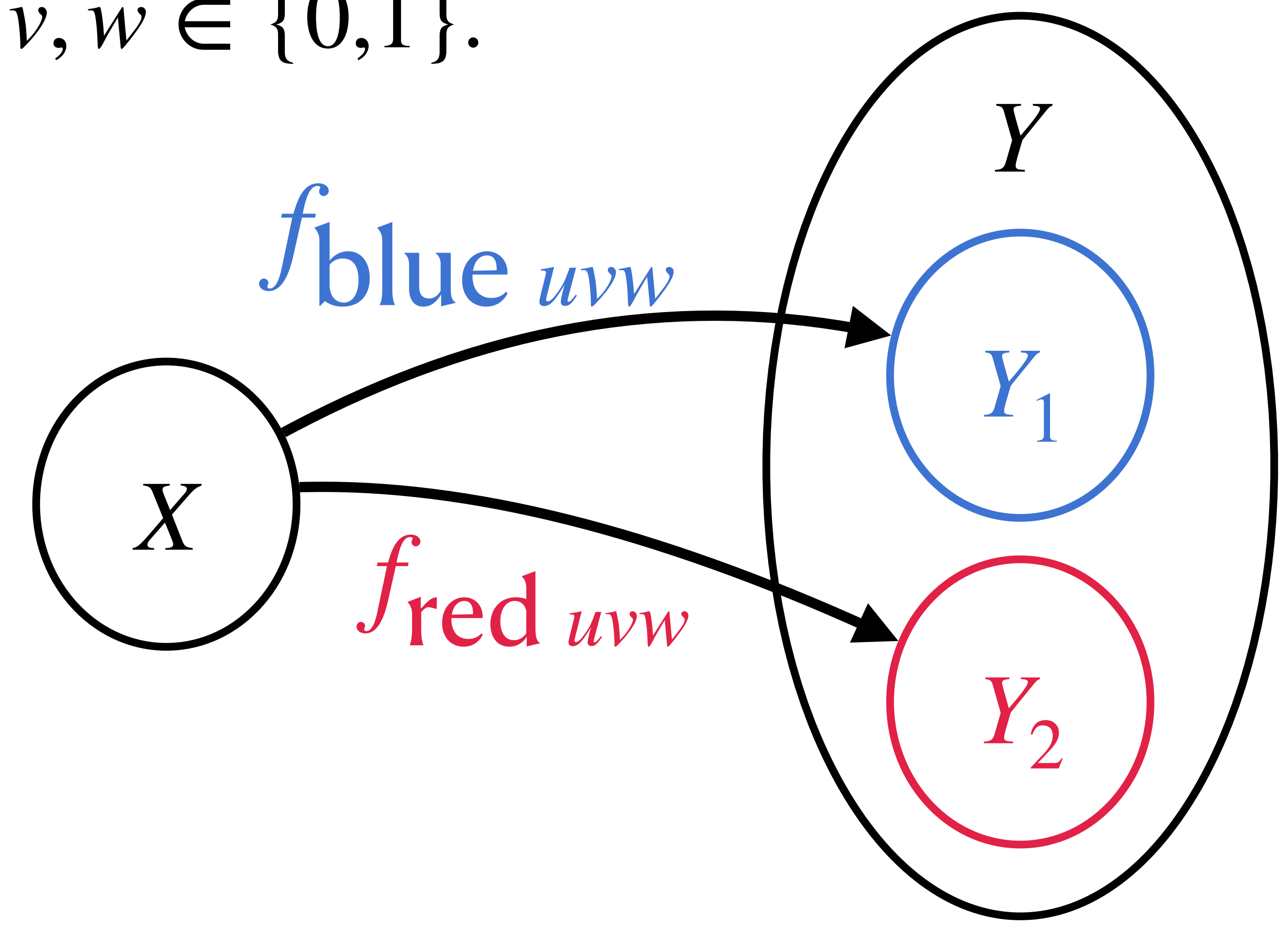
Lemma [GV24]. Dual-mode trapdoor functions $\Longrightarrow$ RSP for $\Big\{$  $_1$ =  ,  $_2$ =  $\Big\}$

If client wants to prepare  $_1$ =  want to project to the blue basis vectors.

$$\frac{1}{2\sqrt{2}}\Big(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big)$$

Dual-mode trapdoor functions

1. Prepare $\sum_{u,v,w\in\{0,1\}} \sum_{x\in X} |uvw\rangle|x\rangle|f_{uvw}(x)\rangle$ for 8 functions $f_{uvw}: X \to Y, u, v, w \in \{0,1\}$.
2. Measure 3rd register, get $y \in Y_1$, $\sum_{\text{blue } u,v,w} |uvw\rangle|x_{uvw}\rangle$.
3. Measure 2nd register in Hadamard basis get rid of x_{uvw}.



Summary of Our Results

Summary of Our Results

Theorem 1. pq. Classical FHE + pq. Dual-mode Trapdoor Functions $\implies$ QFHE.

Summary of Our Results

Theorem 1. pq. Classical FHE + pq. Dual-mode Trapdoor Functions $\implies$ QFHE.

Theorem 2. Group actions $\implies$ pq. Dual-mode Trapdoor Functions.

Summary of Our Results

Theorem 1. pq. Classical FHE + pq. Dual-mode Trapdoor Functions $\implies$ QFHE.

Theorem 2. Group actions $\implies$ pq. Dual-mode Trapdoor Functions.

Corollary 3. pq. IO + Group actions $\implies$ QFHE.

Summary of Our Results

Theorem 1. pq. Classical FHE + pq. Dual-mode Trapdoor Functions $\implies$ QFHE.

Theorem 2. Group actions $\implies$ pq. Dual-mode Trapdoor Functions.

Corollary 3. pq. IO + Group actions $\implies$ QFHE.

Theorem 1. ++ pq. Dual-mode Trapdoor Functions $\implies$ Non-compact QFHE.

Summary of Our Results

Theorem 1. pq. Classical FHE + pq. Dual-mode Trapdoor Functions $\implies$ QFHE.

Theorem 2. Group actions $\implies$ pq. Dual-mode Trapdoor Functions.

Corollary 3. pq. IO + Group actions $\implies$ QFHE.

Theorem 1. ++ pq. Dual-mode Trapdoor Functions $\implies$ Non-compact QFHE.

Corollary 4. Group actions $\implies$ Non-compact QFHE.

Open Question #1

- Is the dream theorem true?

Dream Theorem. Any post-quantum classical FHE $\Longrightarrow$ QFHE.

Open Question #2

Non-compact QFHE from minimal assumptions

- We don’t need classical FHE
- *Our work*: Dual-mode trapdoor functions suffice
- Information-theoretic security unlikely [Morimae-Nishimura-Takeuchi-Tani’18, Aaronson-Cojocaru-Gheorghiu-Kashefi’19]
- Can you construct non-compact QFHE from one-way functions?