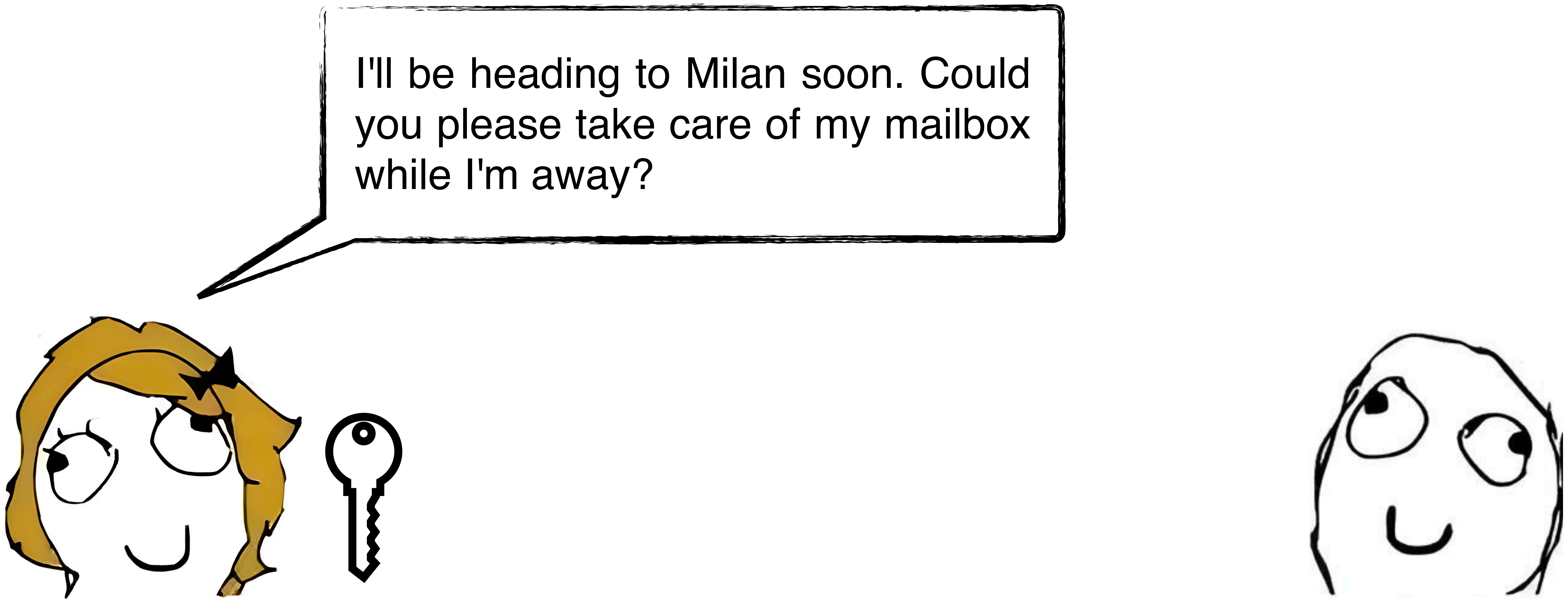
Quantum Key-Revocable Dual-Regev Encryption, Revisited

Prabhanjan Ananth
UCSB

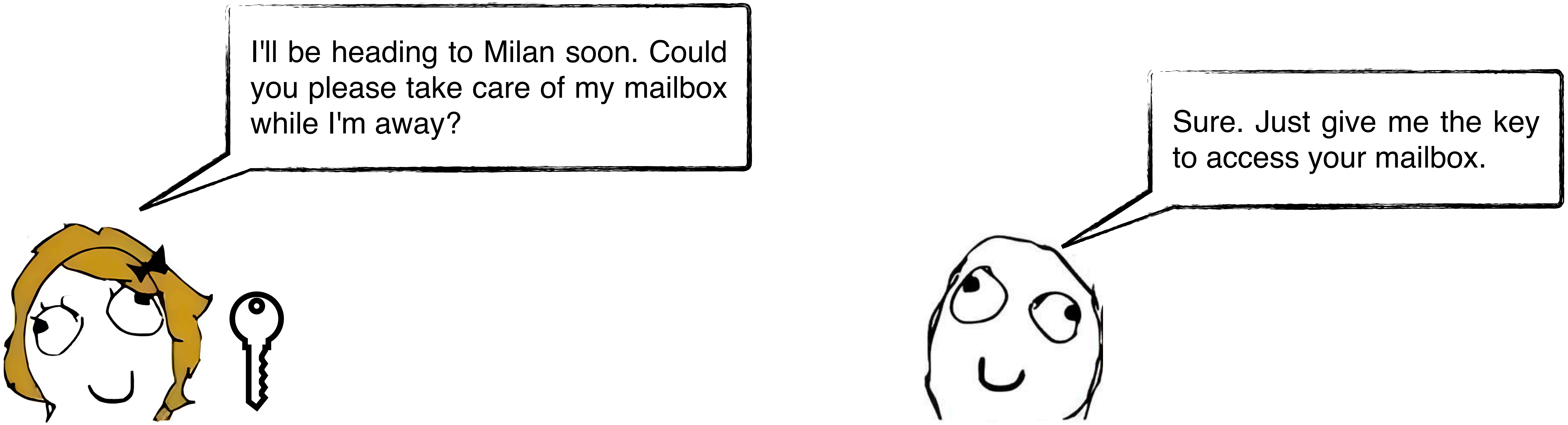
Zihan Hu
EPFL

Zikuan Huang
Tsinghua University

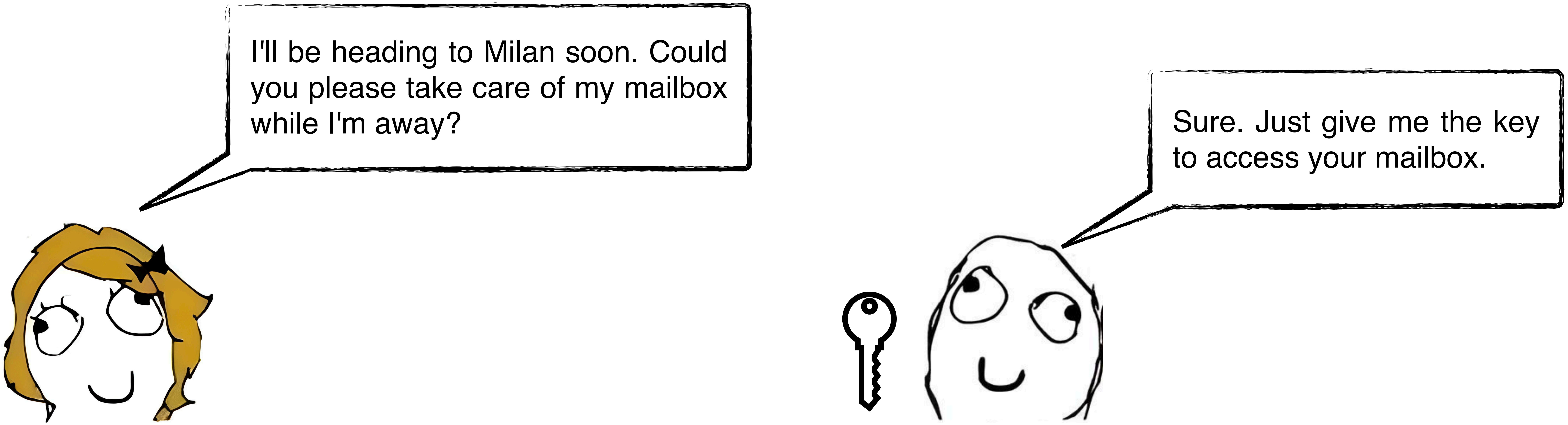
Sometimes we want to delegate and then revoke…



Sometimes we want to delegate and then revoke…

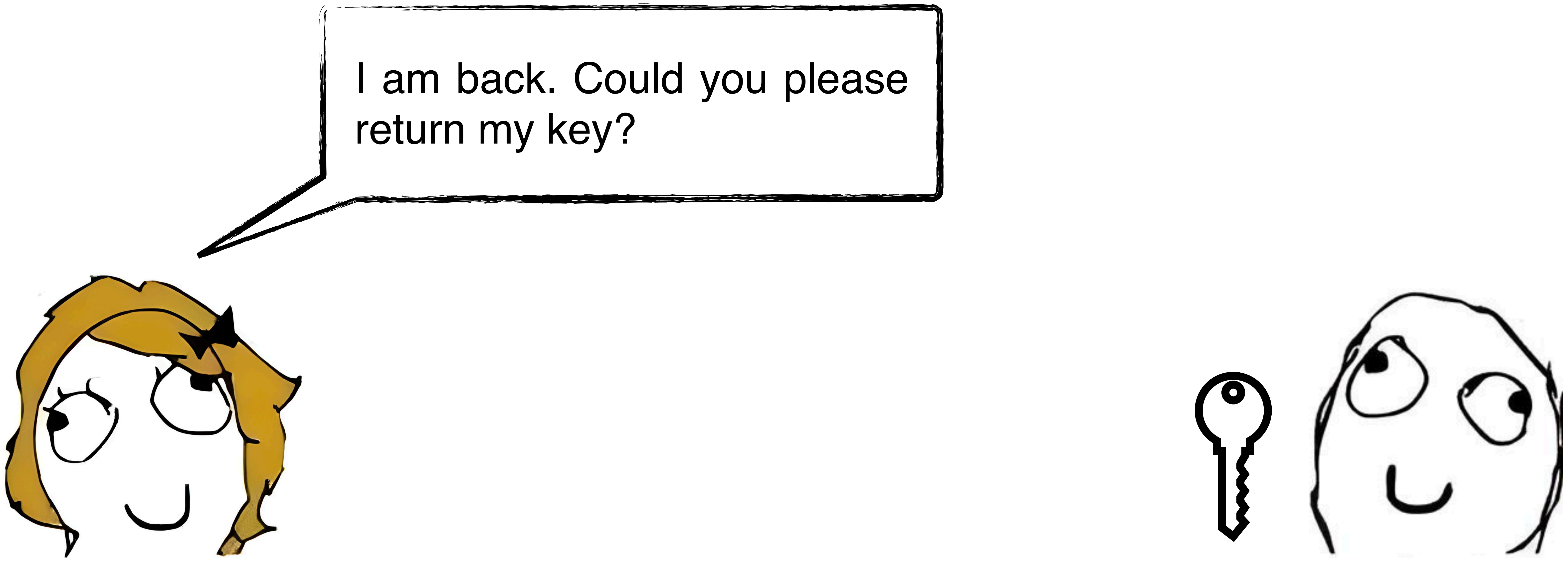


Sometimes we want to delegate and then revoke…



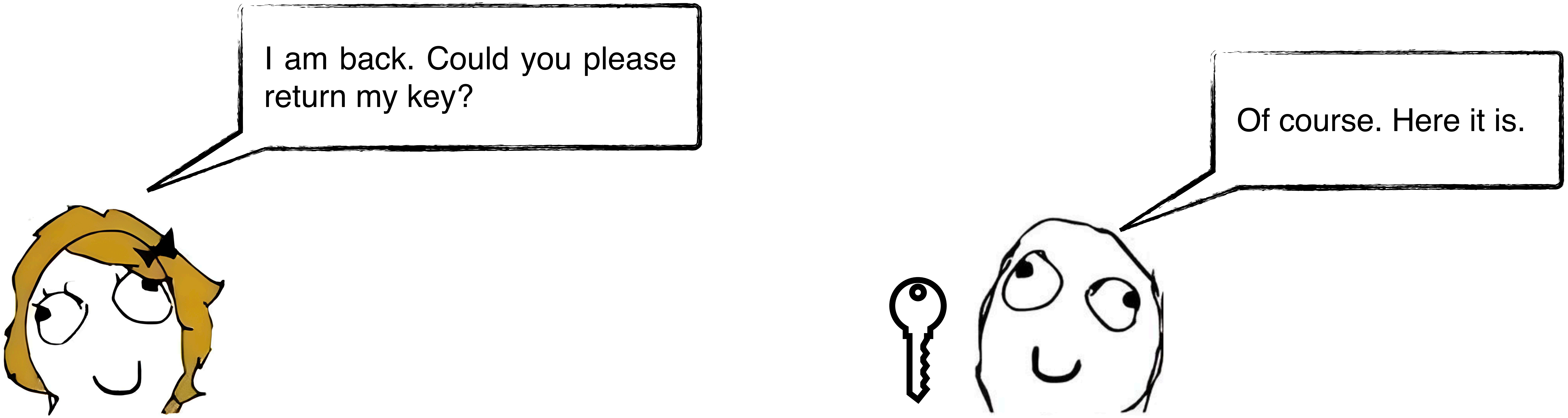
Sometimes we want to delegate and then revoke…

After a week…



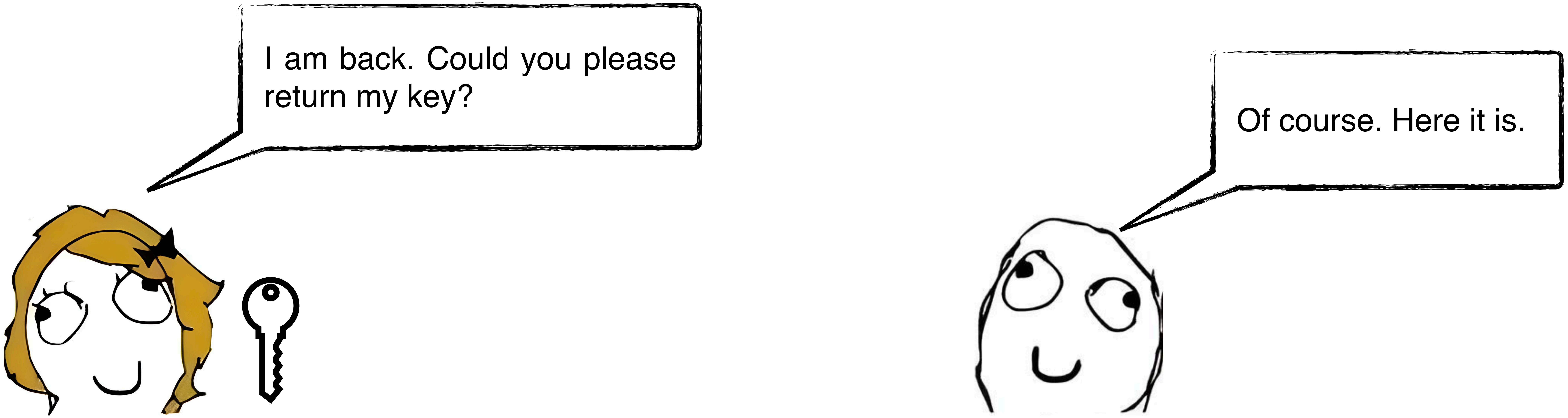
Sometimes we want to delegate and then revoke…

After a week…



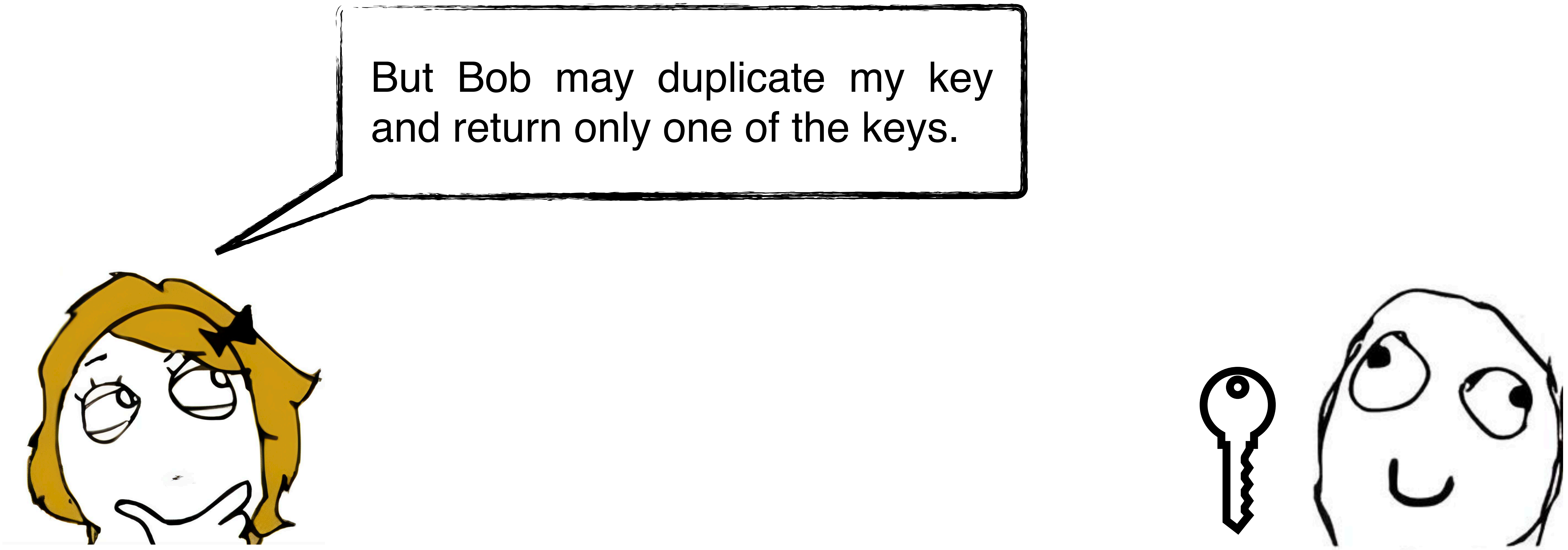
Sometimes we want to delegate and then revoke…

After a week…



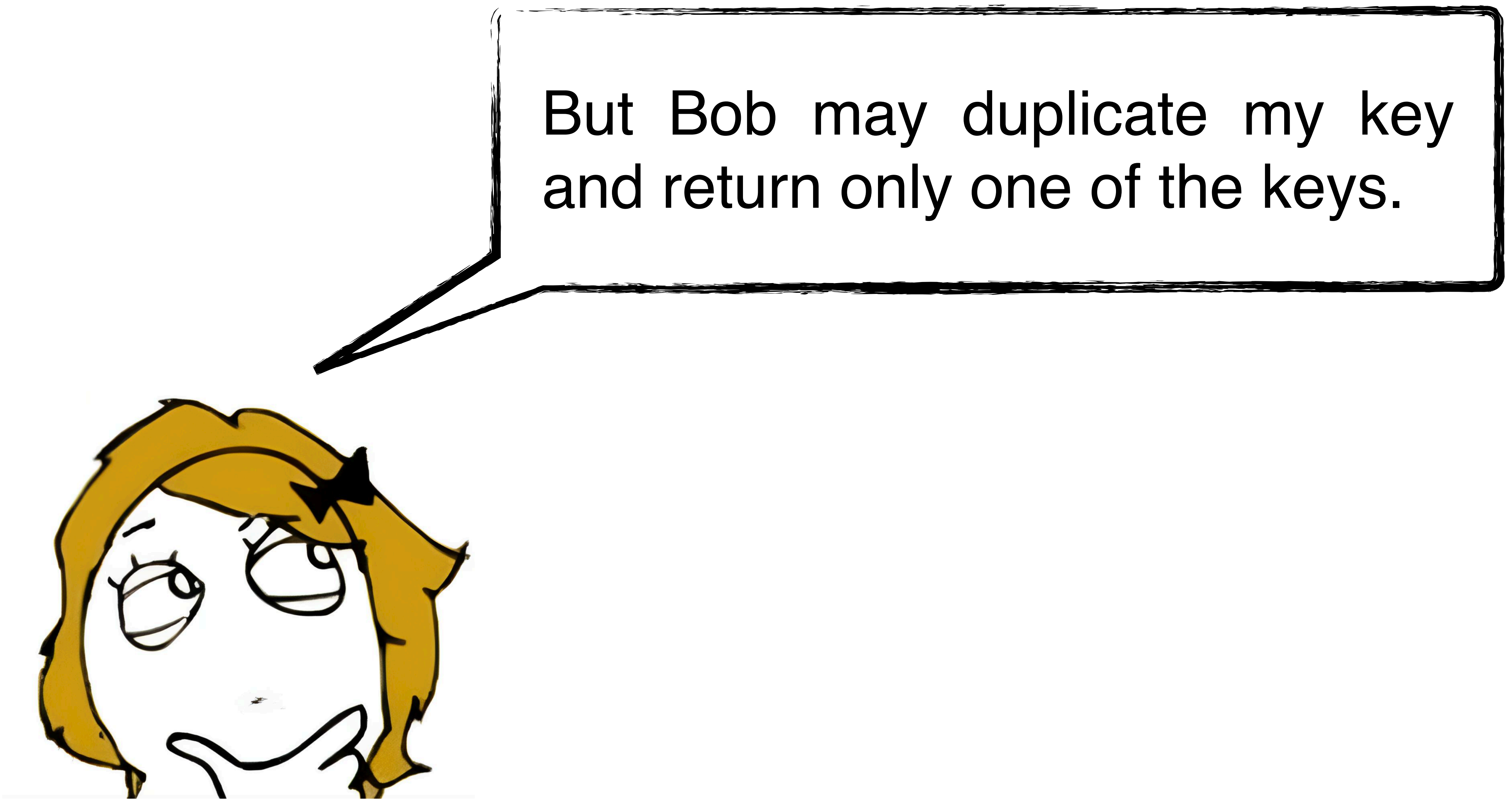
Sometimes we want to delegate and then revoke…

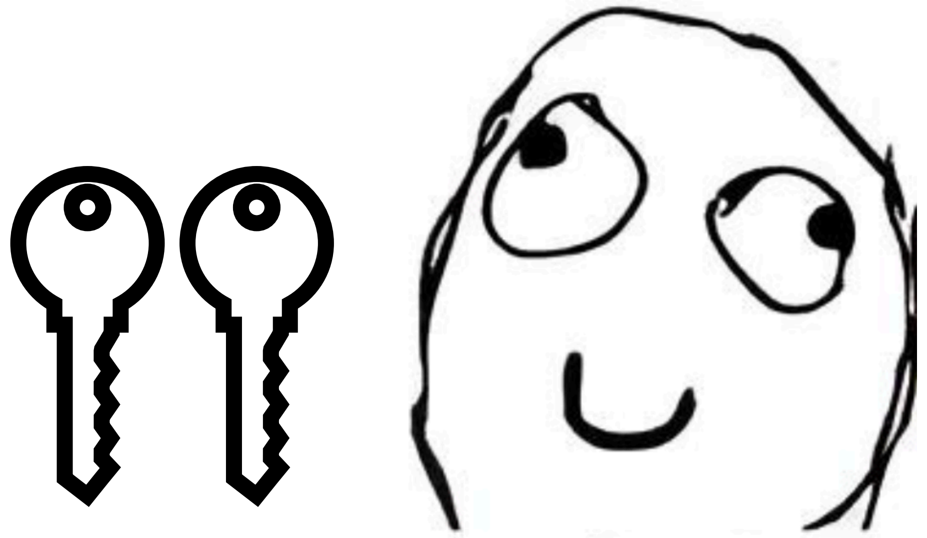
After a week…

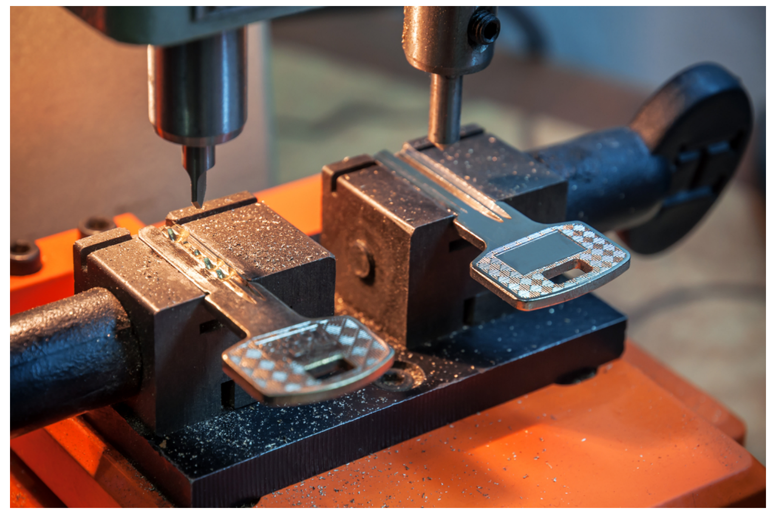


Sometimes we want to delegate and then revoke…

After a week…

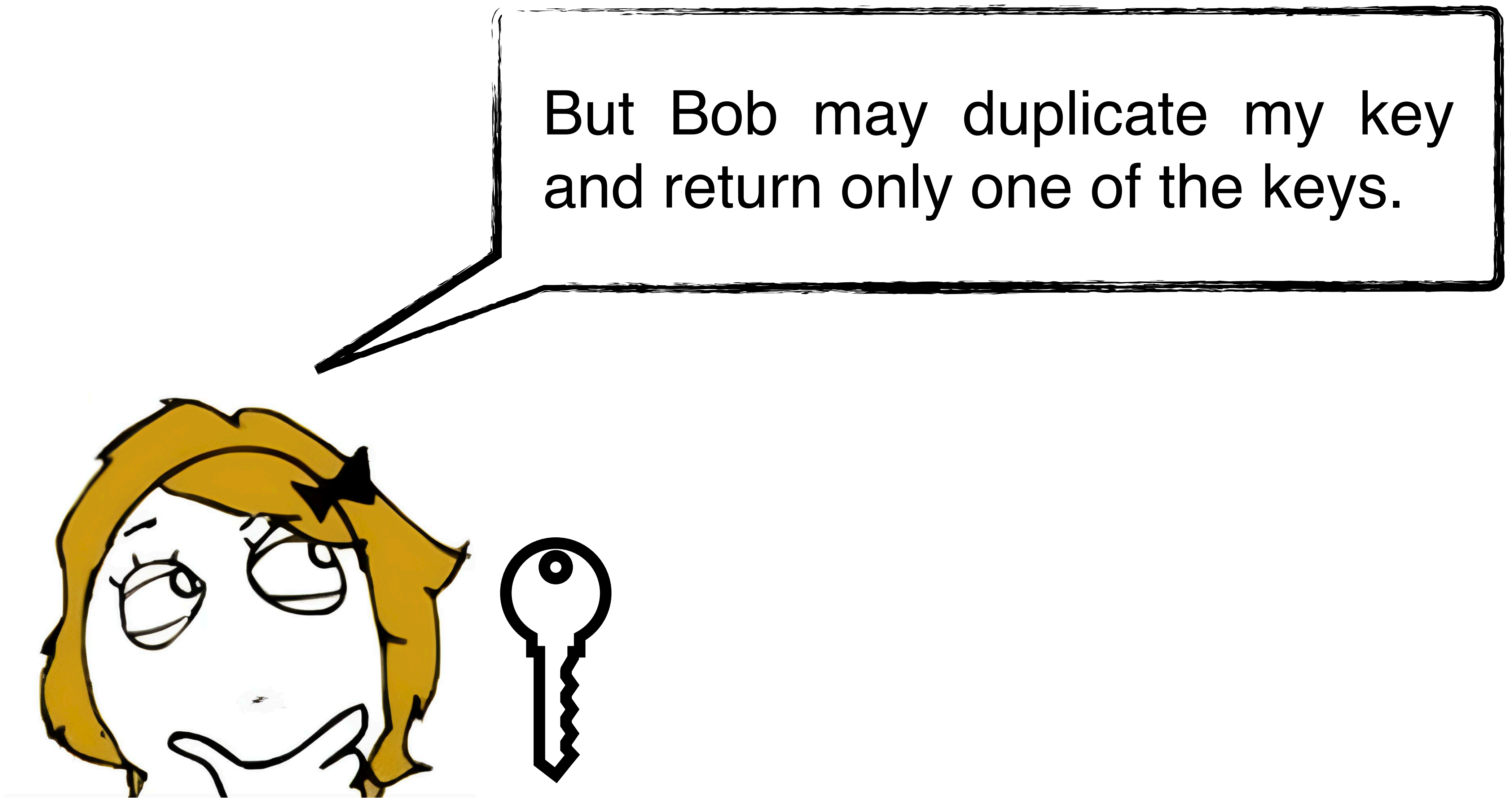




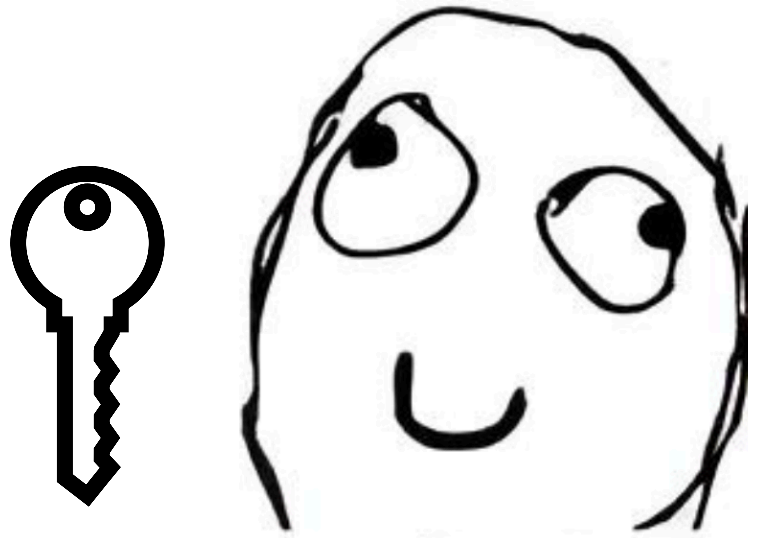


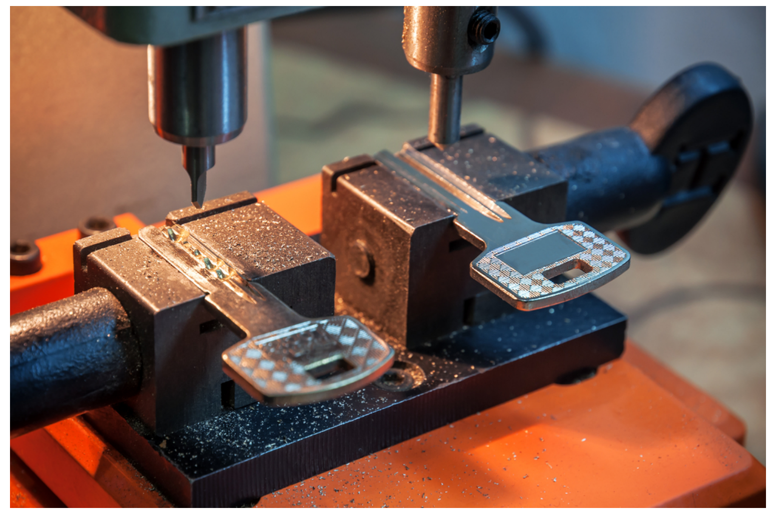
Sometimes we want to delegate and then revoke…

After a week…



Bob may have access to Alice's mailbox after returning one of the keys.



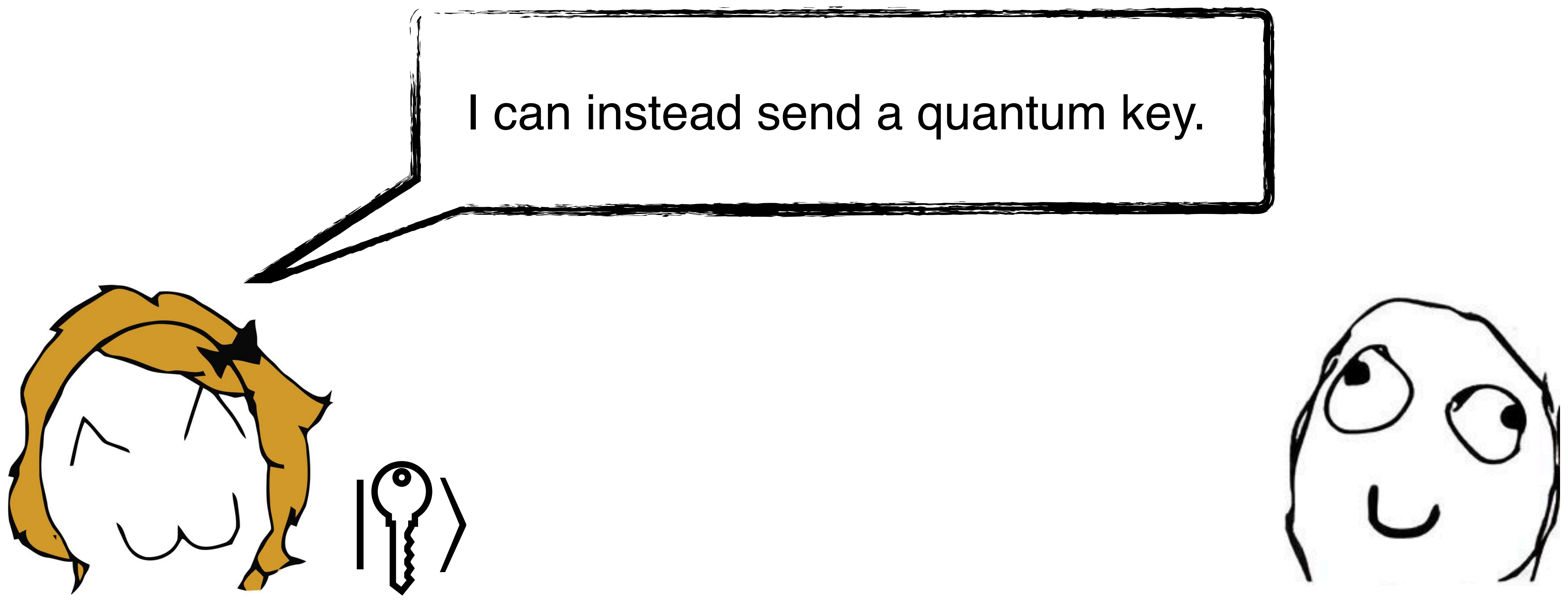


What is revocable cryptography?

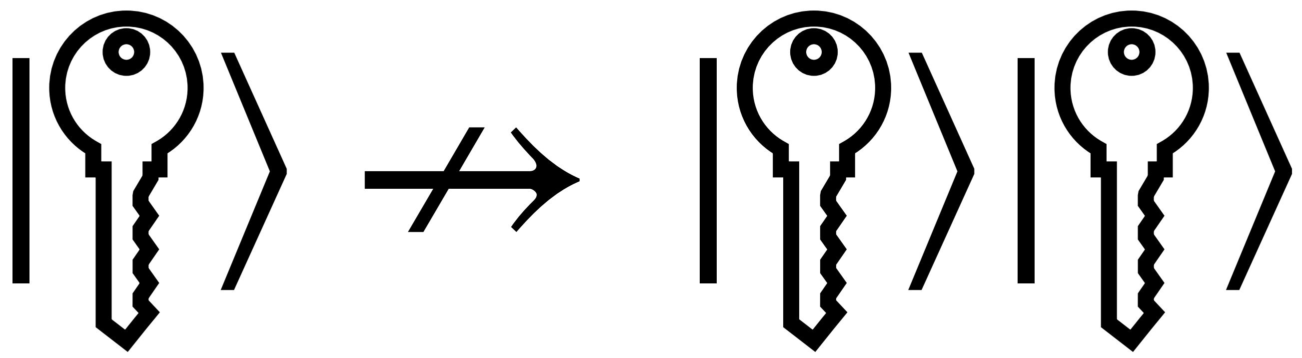
Leverage the no-cloning principle of quantum mechanics to delegate and revoke cryptographic capabilities enabled by secret keys.

What is revocable cryptography?

Leverage the no-cloning principle of quantum mechanics to delegate and revoke cryptographic capabilities enabled by secret keys.

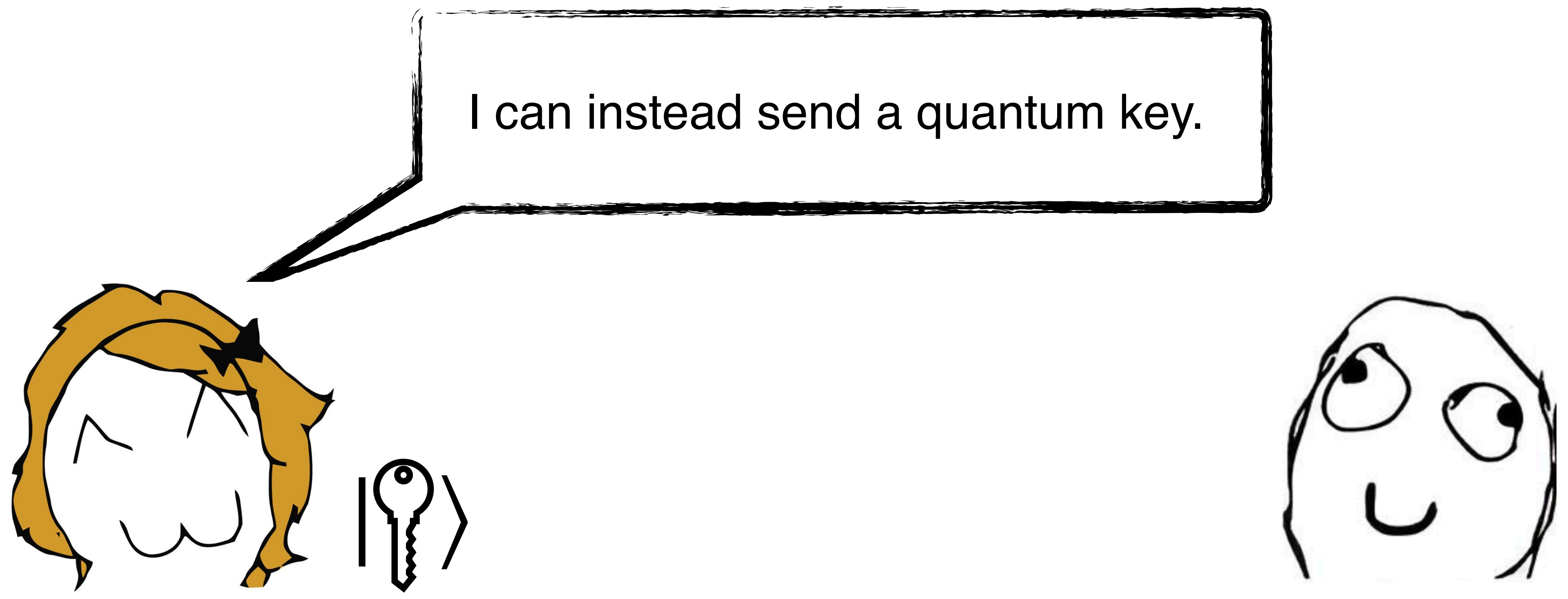


No-Cloning Theorem:



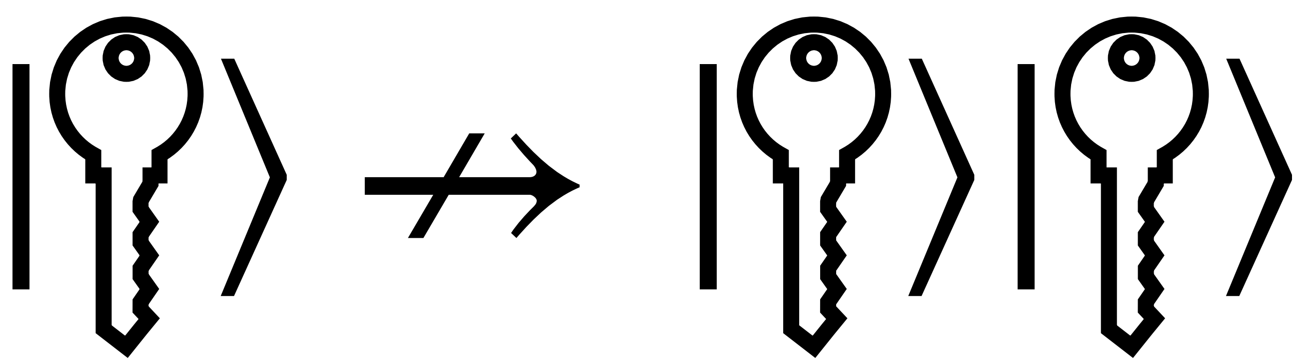
What is revocable cryptography?

Leverage the no-cloning principle of quantum mechanics to delegate and revoke cryptographic capabilities enabled by secret keys.



- It’s weaker than copy-protection [Aar09], yet meaningful, and can be based on weaker assumptions
- Unlike cryptography with certified deletion [BI20, HMNY21, BK22], an honest user is supposed to return the original quantum key for revocation

No-Cloning Theorem:



What is revocable cryptography?

Leverage the no-cloning principle of quantum mechanics to delegate and revoke cryptographic capabilities enabled by secret keys.

Correctness:

(1) with $|$$\rangle$, Bob has the cryptographic capabilities

(2) honest Bob can pass the check Revoke

Security:

After sending a state that passes the check Revoke, Bob no longer has the cryptographic capabilities

What is revocable cryptography?

Leverage the no-cloning principle of quantum mechanics to delegate and revoke cryptographic capabilities enabled by secret keys.

Correctness:

(1) with |⟩, Bob has the cryptographic capabilities

(2) honest Bob can pass the check Revoke

Security:

After sending a state that passes the check Revoke, Bob no longer has the cryptographic capabilities

Revocable public-key encryption

Revocable FHE

Revocable PRF

...

What is revocable cryptography?

Leverage the no-cloning principle of quantum mechanics to delegate and revoke cryptographic capabilities enabled by secret keys.

Correctness:

(1) with $|$$\rangle$, Bob has the cryptographic capabilities

(2) honest Bob can pass the check Revoke

Security:

After sending a state that passes the check Revoke, Bob no longer has the cryptographic capabilities

The ability to decrypt

Revocable public-key encryption

Revocable FHE

The ability to decrypt

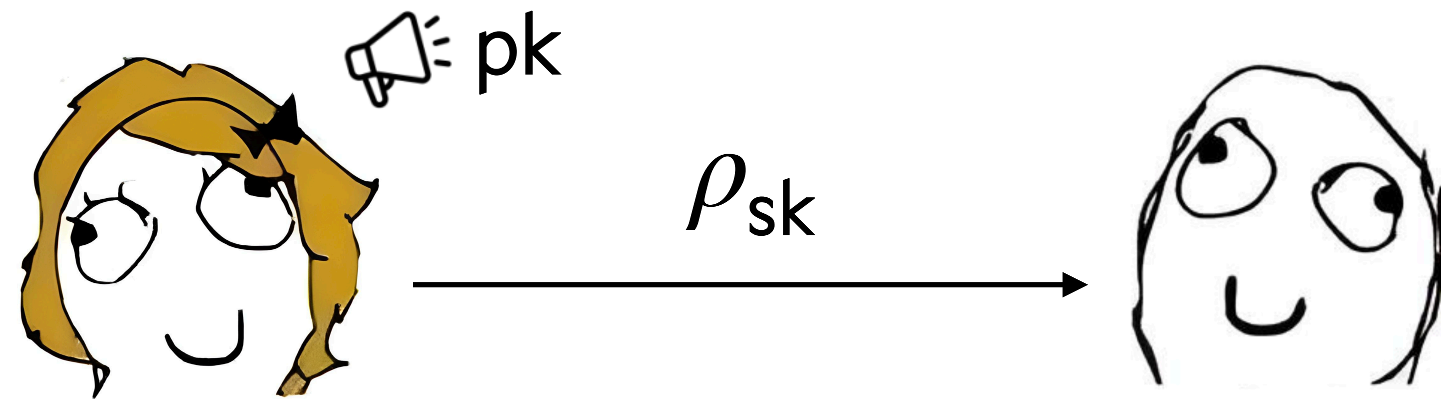
Revocable PRF

The ability to evaluate

...

Revocable PKE: Syntax

Leverage quantum mechanics to delegate and revoke the ability to decrypt.



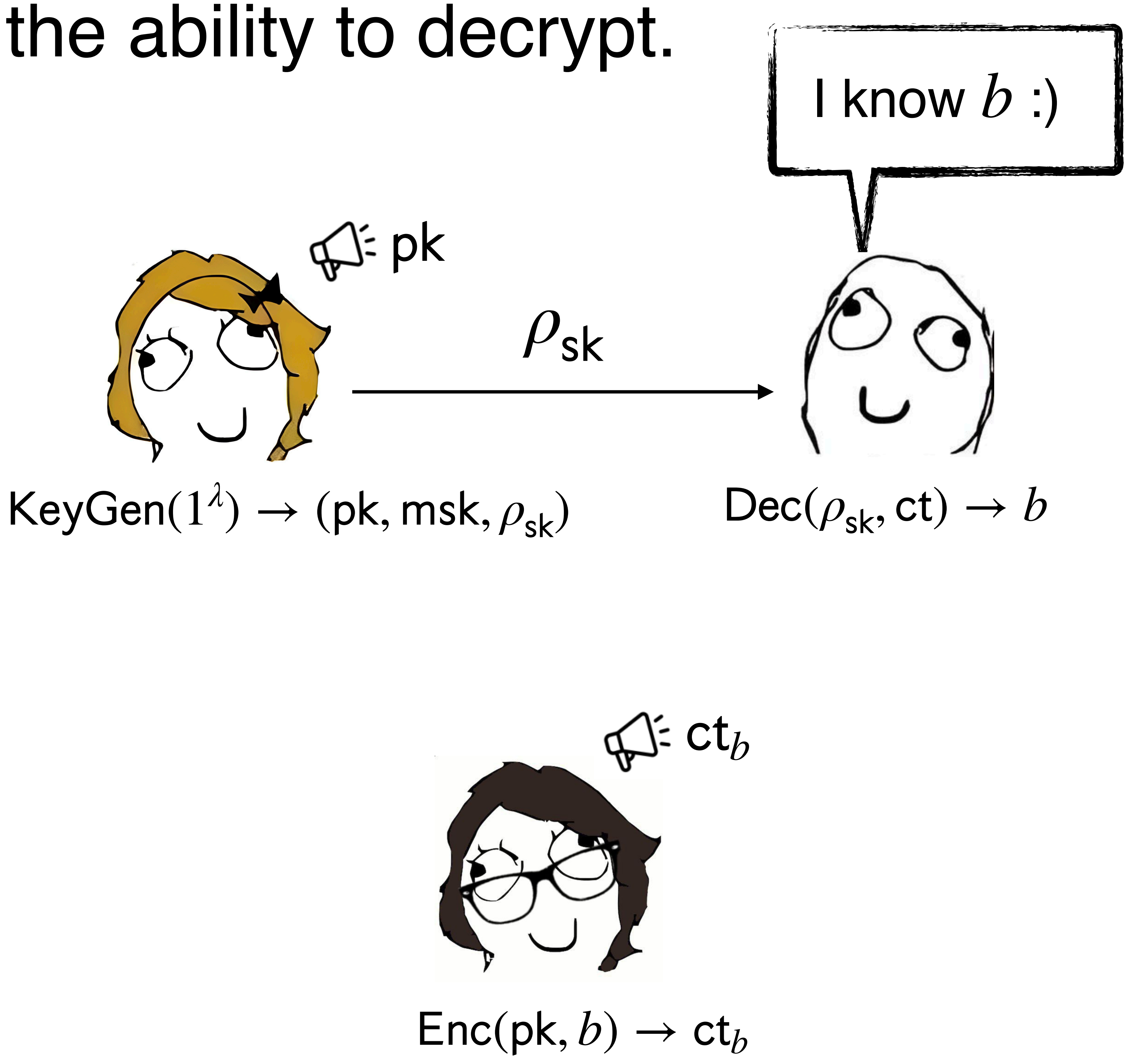
$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk}, \rho_{\mathsf{sk}})$

Revocable PKE: Syntax

Leverage quantum mechanics to delegate and revoke the ability to decrypt.

Correctness:

(1) with , Bob can decrypt



It suffices to consider encryption for a bit $b \in \{0,1\}$.

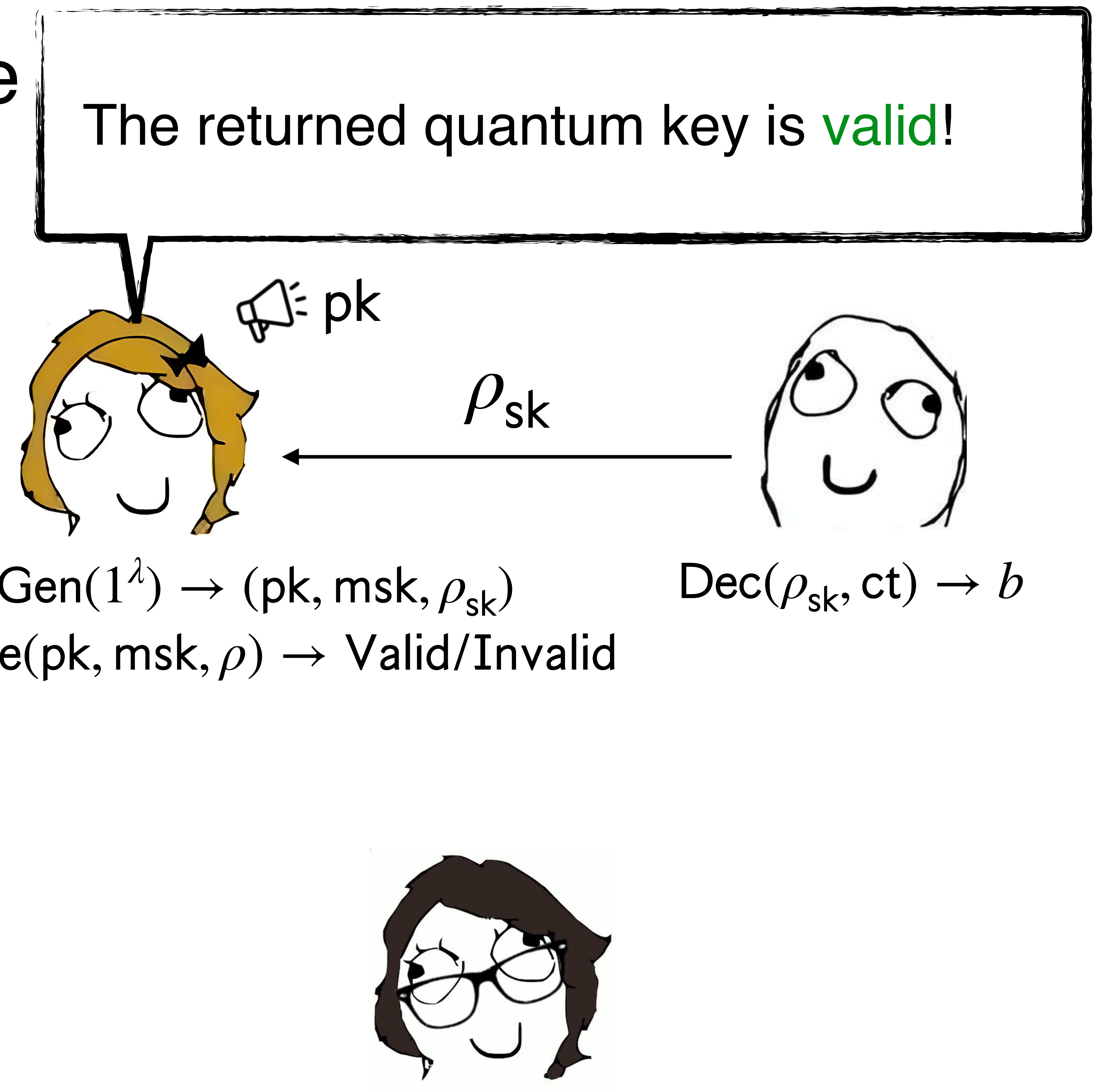
Revocable PKE: Syntax

Leverage quantum mechanics to delegate and revoke the

Correctness:

(1) with  , Bob can decrypt

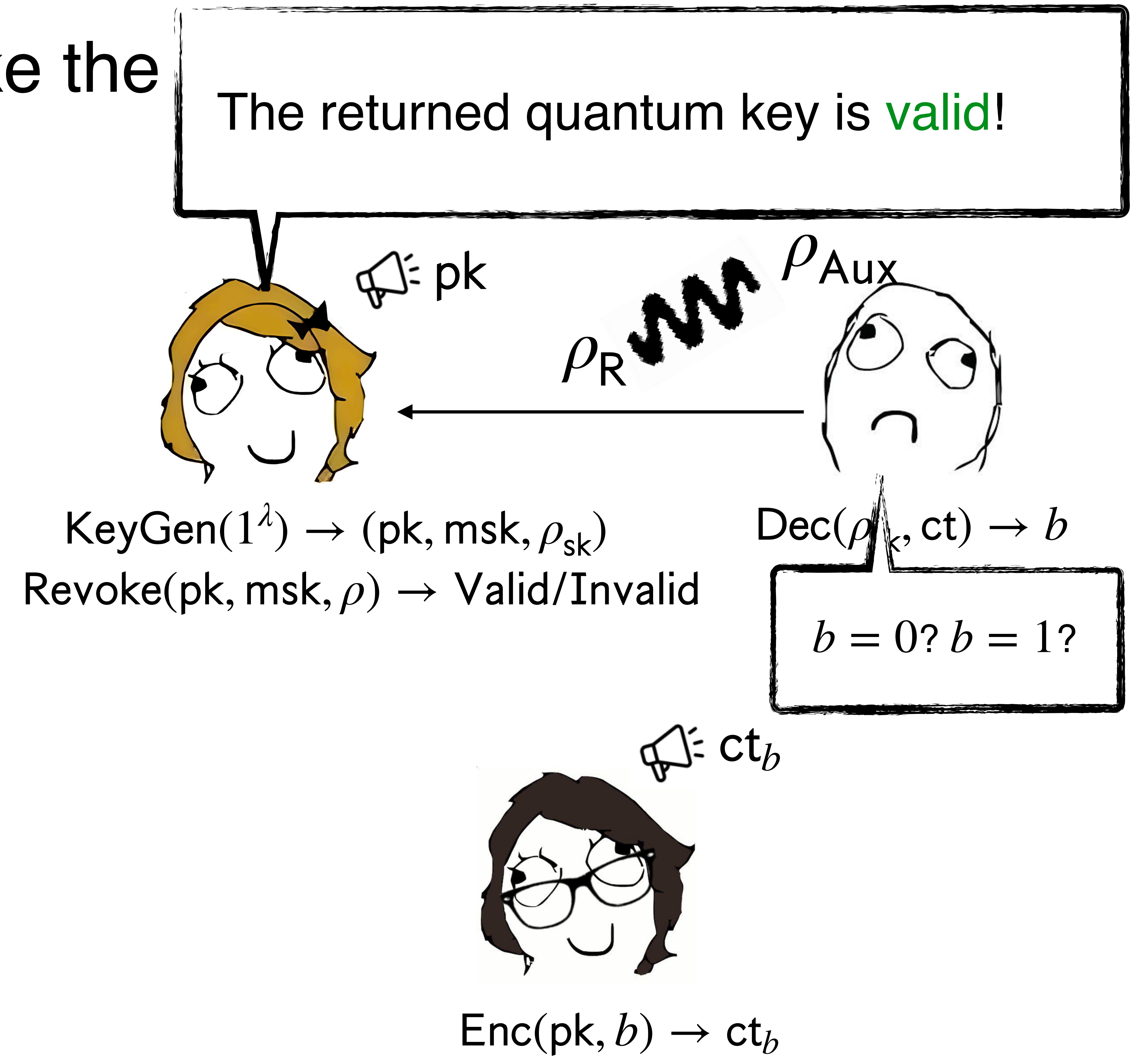
(2) honest Bob can pass the check Revoke



It suffices to consider encryption for a bit $b \in \{0,1\}$.

Revocable PKE: Syntax

Leverage quantum mechanics to delegate and revoke the



Correctness:

(1) with |  ⟩, Bob can decrypt

(2) honest Bob can pass the check Revoke

Security:

After sending a state that passes the check Revoke, polynomial-time Bob can no longer distinguish encryption of 0 and encryption of 1

It suffices to consider encryption for a bit $b \in \{0,1\}$.

Prior work

- Assuming post-quantum PKE, there exists a revocable PKE scheme [AKN+23]
- Assuming simultaneous dual-Regev conjecture, the dual-Regev PKE scheme is revocable [APV23]
- Assuming post-quantum sub-exponential hardness of LWE, there exists a revocable PKE scheme with classical revocation [CGJL23]

Prior work

- Assuming post-quantum PKE, there exists a revocable PKE scheme [AKN+23]
- Assuming simultaneous dual-Regev conjecture, the dual-Regev PKE scheme is revocable [APV23]
- Assuming post-quantum sub-exponential hardness of LWE, there exists a revocable PKE scheme with classical revocation [CGJL23]

[APV23]: Can we prove that the dual-Regev PKE scheme is revocable from LWE?

Prior work

- Assuming post-quantum PKE, there exists a revocable PKE scheme [AKN+23]
- Assuming simultaneous dual-Regev conjecture, the dual-Regev PKE scheme is revocable [APV23]
- Assuming post-quantum sub-exponential hardness of LWE, there exists a revocable PKE scheme with classical revocation [CGJL23]

[APV23]: Can we prove that the dual-Regev PKE scheme is revocable from LWE?

Why do we care about the dual-Regev PKE scheme?

(1) [APV23] gave many reductions from revocable dual-Regev PKE scheme!

(2) It's a textbook PKE, and may inspire other protocols with similar structures.

Our work

Assuming post-quantum polynomial hardness of LWE over sub-exponential modulus,

- The dual-Regev PKE scheme (the construction in [APV23]) is revocable

+ the results in [APV23]

Assuming post-quantum polynomial hardness of LWE over sub-exponential modulus,

- The dual-Regev PKE scheme has classical revocation
- There exists revocable FHE with quantum/classical revocation
- There exists revocable PRF with quantum/classical revocation

Our work

Assuming post-quantum polynomial hardness of LWE over sub-exponential modulus,

- The dual-Regev PKE scheme (the construction in [APV23]) is revocable

+ the results in [APV23]

Assuming post-quantum polynomial hardness of LWE over sub-exponential modulus,

- The dual-Regev PKE sche
- There exists revocable FHE with quantum/classical revocation
- There exists revocable PRF with quantum/classical revocation

The first revocable PRF from concrete assumptions

Recall: Dual-Regev PKE

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

To encrypt: $\mathsf{Enc}(\mathsf{pk}, b) = (\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + b\left\lfloor\frac{q}{2}\right\rfloor + e')$

<u>The classical decryption key</u>:

A short preimage $\mathbf{x}$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

To decrypt: Notice that $\mathbf{s}^T\mathbf{y} + b\left\lfloor\frac{q}{2}\right\rfloor + e' - \left(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T\right)\mathbf{x} \approx b\left\lfloor\frac{q}{2}\right\rfloor$

Recall: Dual-Regev PKE

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

To encrypt: $\mathsf{Enc}(\mathsf{pk}, b) = (\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + b\left\lfloor \frac{q}{2} \right\rfloor + e')$

<u>The classical decryption key</u>:

A short preimage $\mathbf{x}$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

<u>The quantum decryption key</u>:

A superposition of short preimages $\mathbf{x}$ such that

$$|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x}) \, | \, \mathbf{x}\rangle$$

To decrypt: Notice that $\mathbf{s}^T\mathbf{y} + b\left\lfloor \frac{q}{2} \right\rfloor + e' - \left(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T\right)\mathbf{x} \approx b\left\lfloor \frac{q}{2} \right\rfloor$

Recall: Dual-Regev PKE

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

To encrypt: $\mathsf{Enc}(\mathsf{pk}, b) = (\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + b\left\lfloor\frac{q}{2}\right\rfloor + e')$

The classical decryption key:

A short preimage $\mathbf{x}$ such that

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

The quantum decryption key:

A superposition of short preimages $\mathbf{x}$ such that

$$|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x})\,|\,\mathbf{x}\rangle$$

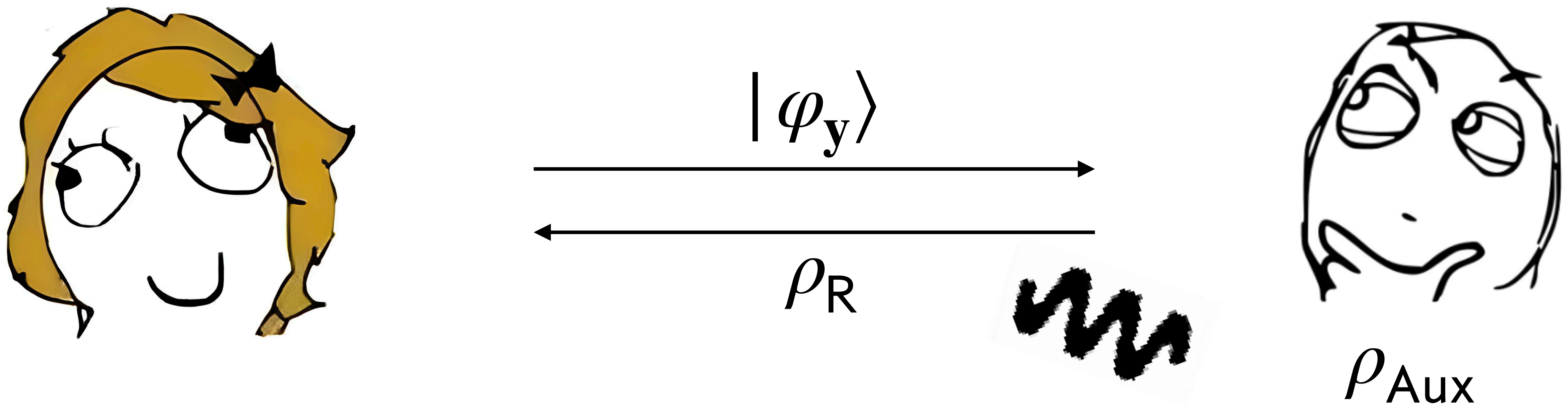
To decrypt: Notice that $\mathbf{s}^T\mathbf{y} + b\left\lfloor\frac{q}{2}\right\rfloor + e' - \left(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T\right)\mathbf{x} \approx b\left\lfloor\frac{q}{2}\right\rfloor$

To revoke: use msk (a short basis of $\mathbf{A}$) to check whether the returned state is $|\varphi_{\mathbf{y}}\rangle$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x}) |\mathbf{x}\rangle$

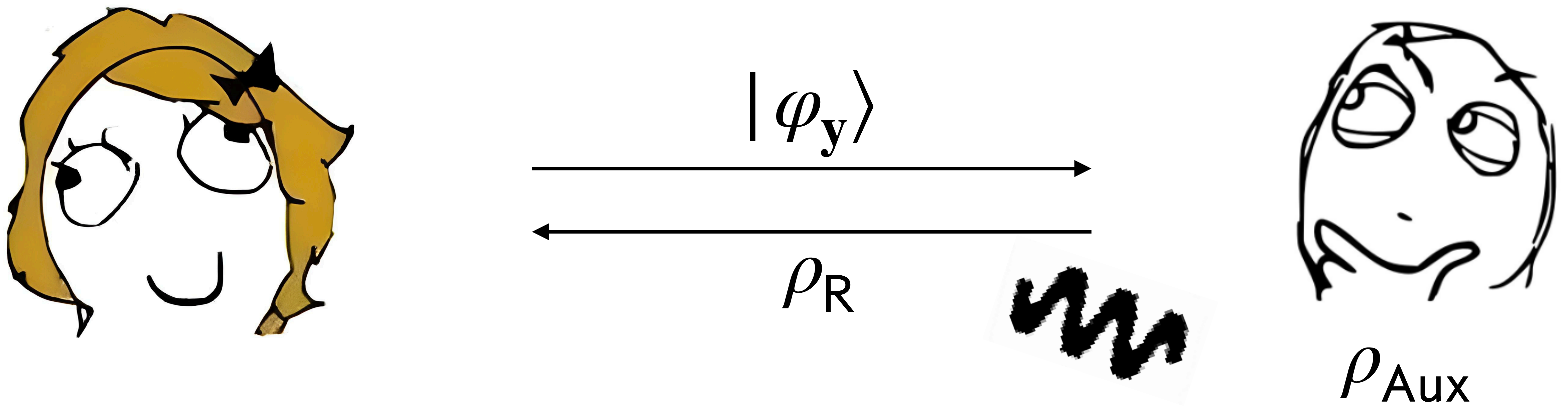


$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x})\,|\,\mathbf{x}\rangle$



$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$

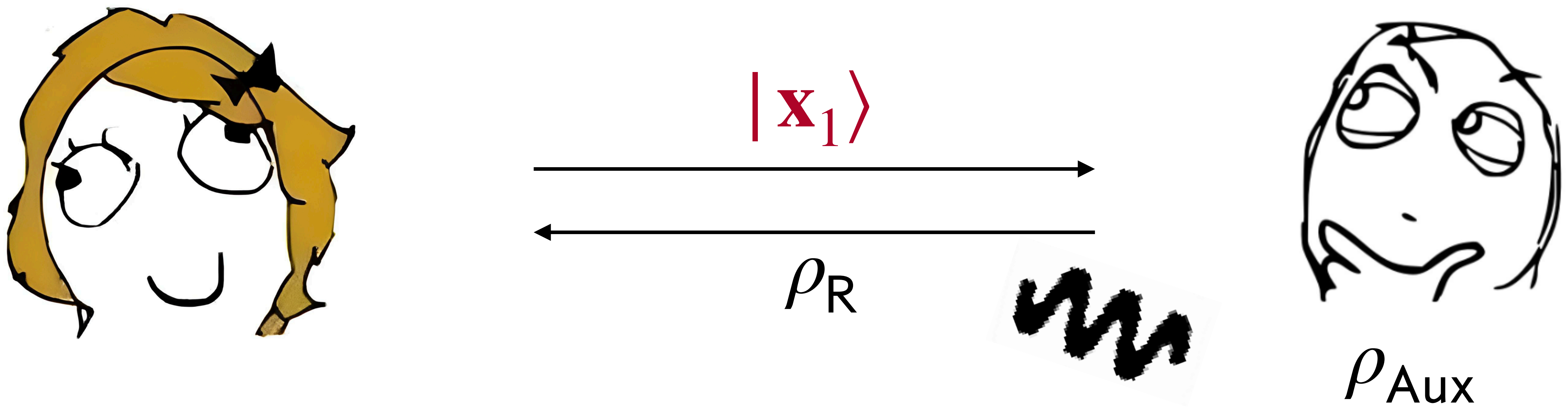
Extract a short preimage $\mathbf{x}_0$ from R and a short preimage $\mathbf{x}_1$ from Aux

Then use $\mathbf{x}_0 - \mathbf{x}_1$ to break SIS!

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x})\,|\mathbf{x}\rangle$

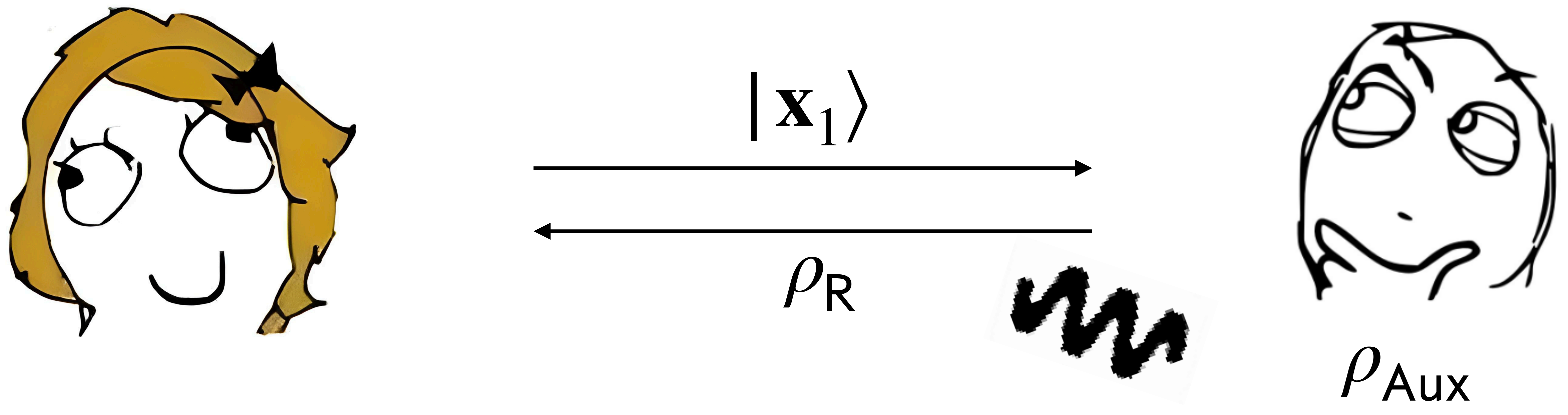


$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{Ax}=\mathbf{y}} \rho_\sigma(\mathbf{x})|\mathbf{x}\rangle$



$(\mathbf{u}, \mathbf{u}^T\mathbf{x}_1 + e')$ vs $(\mathbf{u}, r)$

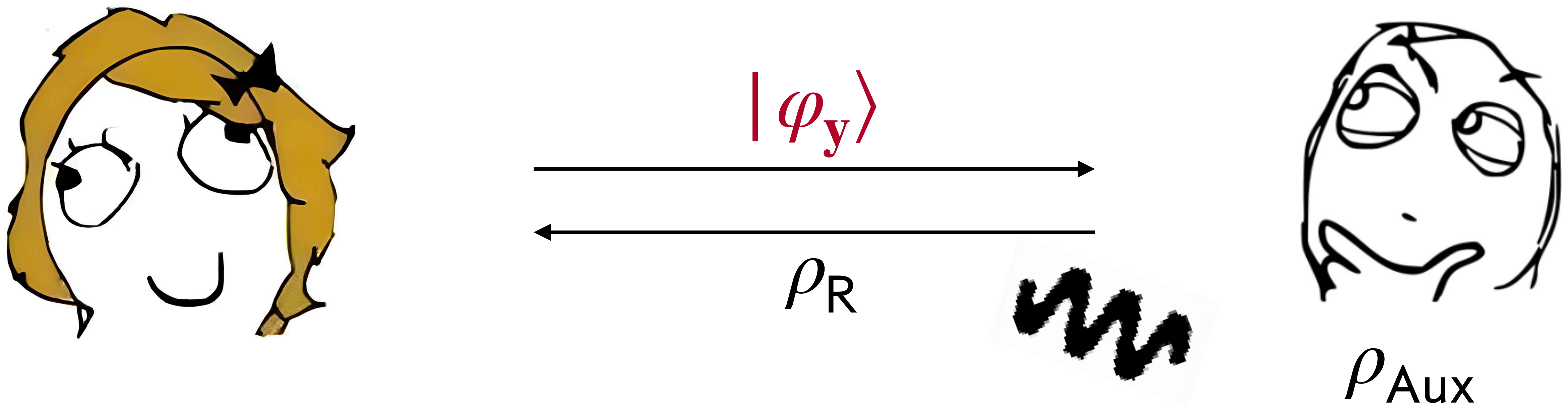
Extract a short preimage $\mathbf{x}_1$

$$\mathbf{Ax}_1 = \mathbf{y}$$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle$



$(\mathbf{u}, \mathbf{u}^T\mathbf{x}_1 + e')$ vs $(\mathbf{u}, r)$

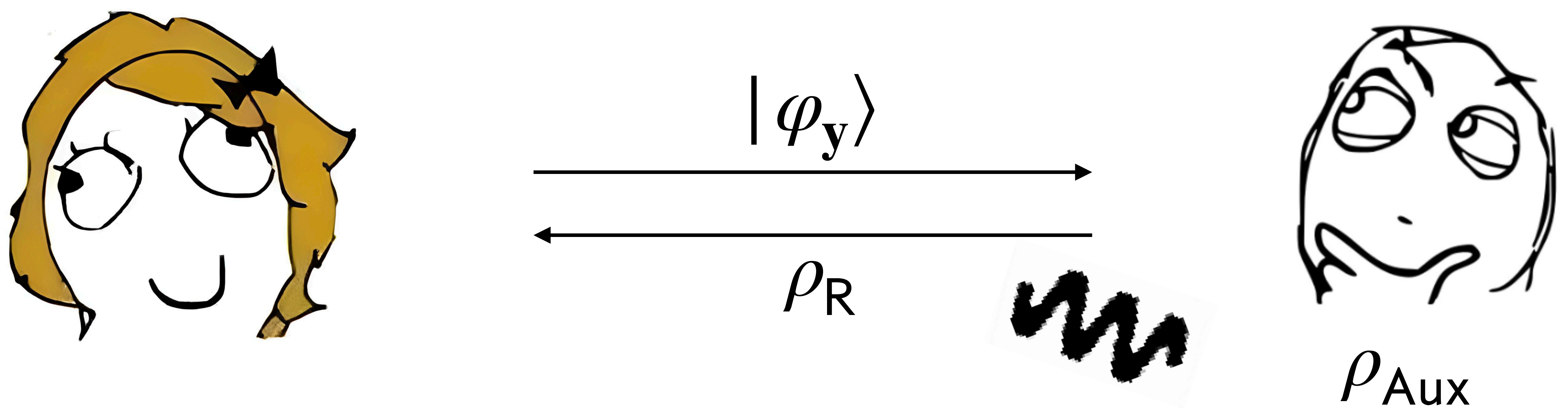
Extract a short preimage $\mathbf{x}_1$

$$\mathbf{A}\mathbf{x}_1 = \mathbf{y}$$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x})\,|\mathbf{x}\rangle$



Revoke passes: $\rho_{\mathrm{R}} \approx |\varphi_{\mathbf{y}}\rangle\langle\varphi_{\mathbf{y}}|$

$(\mathbf{u}, \mathbf{u}^T\mathbf{x}_1 + e')$ vs $(\mathbf{u}, r)$

Computational Measurement => a short preimage $\mathbf{x}_0$

$$\mathbf{A}\mathbf{x}_0 = \mathbf{y}$$

Extract a short preimage $\mathbf{x}_1$

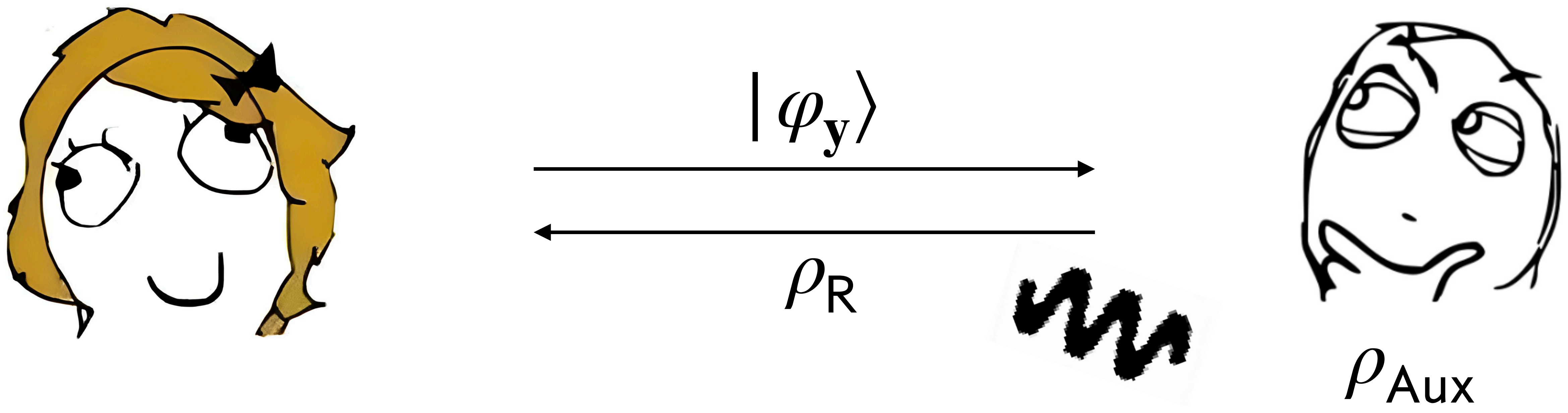
$$\mathbf{A}\mathbf{x}_1 = \mathbf{y}$$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x}) |\mathbf{x}\rangle$

Challenge: get $\mathbf{x}_0$ and $\mathbf{x}_1$ simultaneously?



Revoke passes: $\rho_{\mathsf{R}} \approx |\varphi_{\mathbf{y}}\rangle\langle\varphi_{\mathbf{y}}|$

$(\mathbf{u}, \mathbf{u}^T\mathbf{x}_1 + e')$ vs $(\mathbf{u}, r)$

Computational Measurement => a short preimage $\mathbf{x}_0$

$$\mathbf{A}\mathbf{x}_0 = \mathbf{y}$$

Extract a short preimage $\mathbf{x}_1$

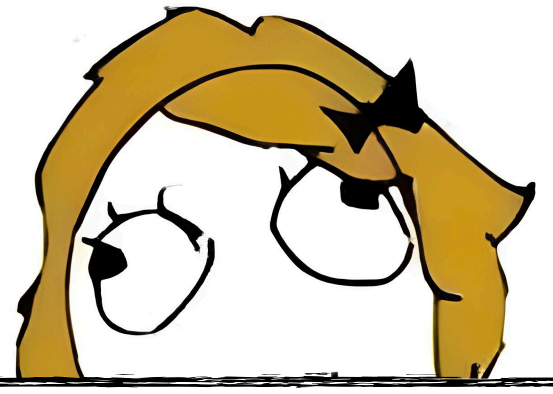
$$\mathbf{A}\mathbf{x}_1 = \mathbf{y}$$

The route in [APV23]

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

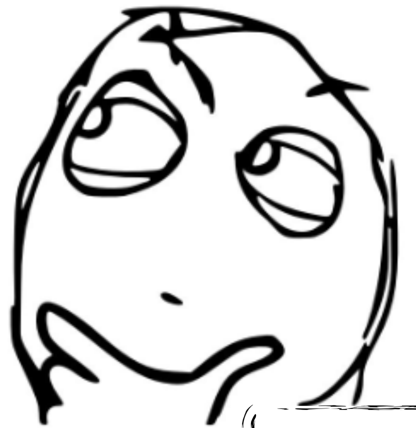
The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x})\,|\,\mathbf{x}\rangle$

Challenge: get $\mathbf{x}_0$ and $\mathbf{x}_1$ simultaneously?



$|\varphi_{\mathbf{y}}\rangle$ →

← ρ_{R}



ρ_{Aux}

Revocation succeeds w.p. $1/\text{poly}(n)$

Extraction succeeds w.p. $1/\text{poly}(q)$

Revoke passes: $\rho_{\text{R}} \approx |\varphi_{\mathbf{y}}\rangle\langle\varphi_{\mathbf{y}}|$

$(\mathbf{u}, \mathbf{u}^T\mathbf{x}_1 + e')$ vs $(\mathbf{u}, r)$

Computational Measurement => a short preimage $\mathbf{x}_0$

$\mathbf{A}\mathbf{x}_0 = \mathbf{y}$

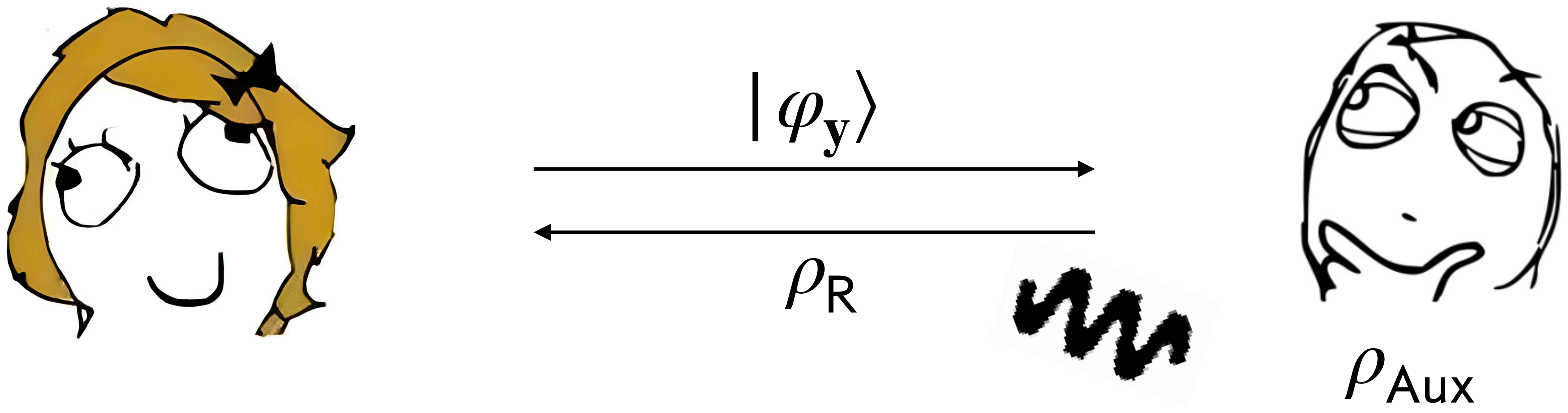
Extract a short preimage $\mathbf{x}_1$

$\mathbf{A}\mathbf{x}_1 = \mathbf{y}$

Our approach: almost perfect extraction

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle$

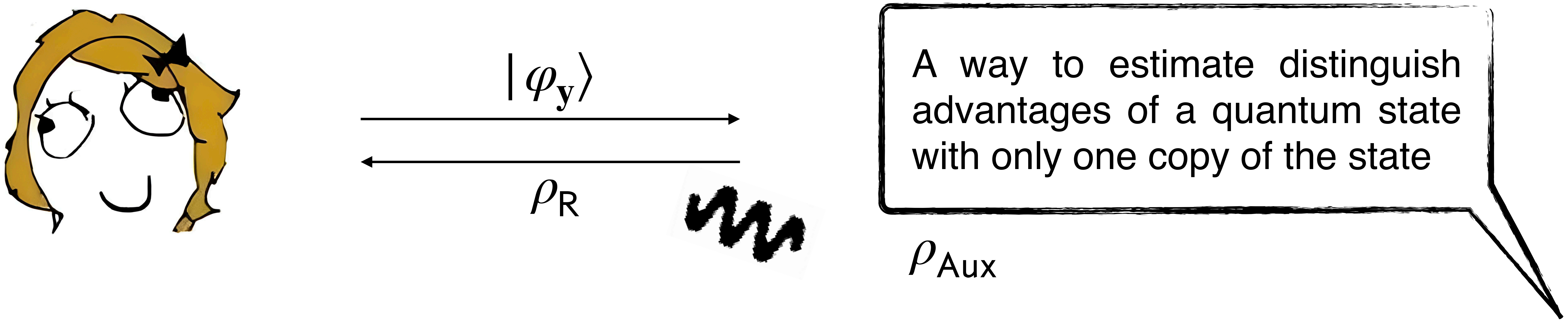


- Test if it is a good distinguisher for $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$ via ATI

Our approach: almost perfect extraction

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x}) |\mathbf{x}\rangle$

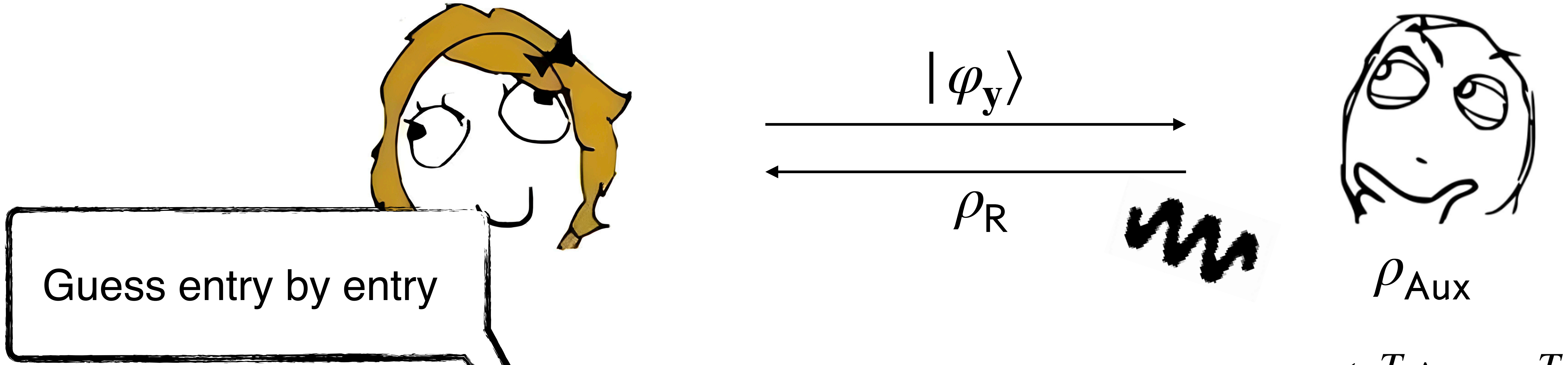


- Test if it is a good distinguisher for $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$ via ATI

Our approach: almost perfect extraction

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x} = \mathbf{y}} \rho_\sigma(\mathbf{x}) \, |\mathbf{x}\rangle$



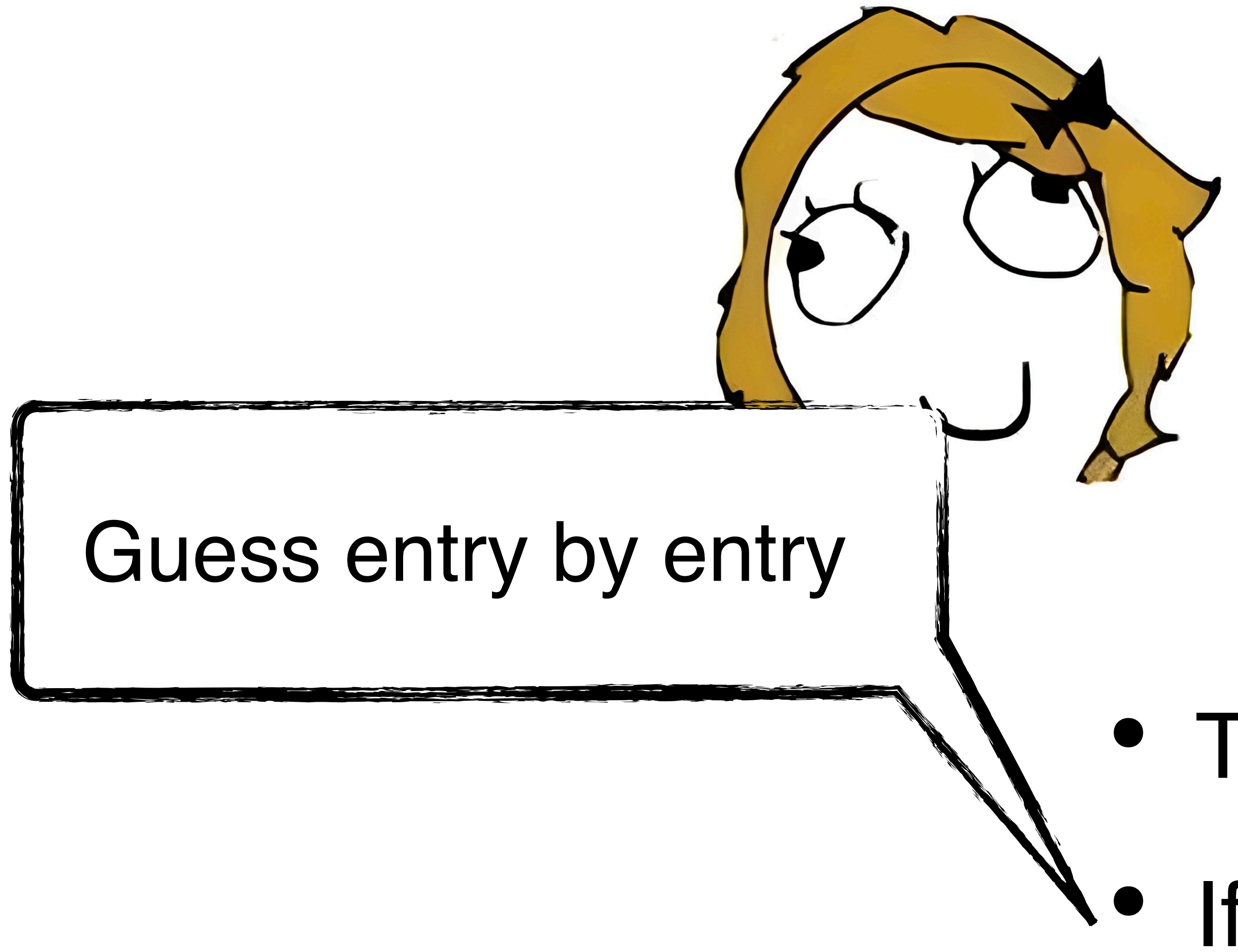
- Test if it is a good distinguisher for $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$ via ATI
- If yes, test it on $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c \cdot g)$ vs $(\mathbf{u}, r)$ for each guess g
  - If the i^{th} entry of $\mathbf{x}_1$ is g, it's a good distinguisher with certainty
  - Otherwise, it's a bad distinguisher with certainty

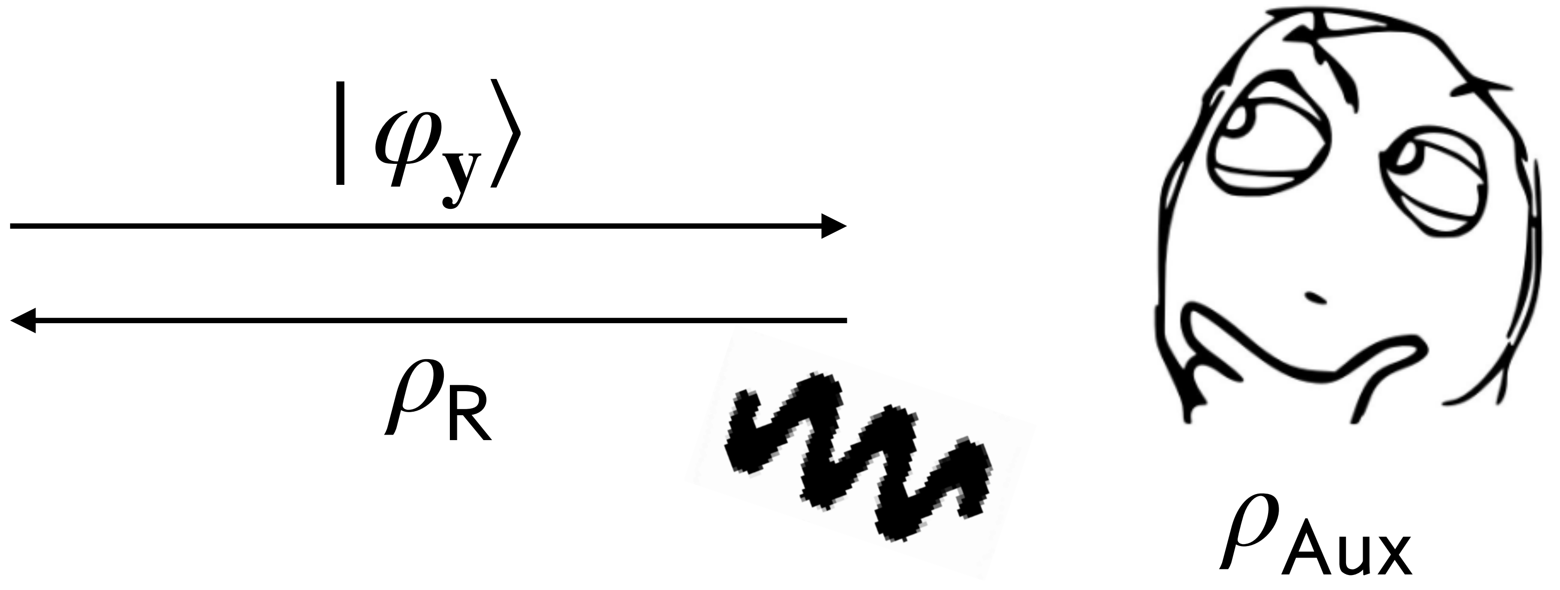
$\mathbf{i}$ is the unit vector where the i^{th} entry is 1.

Our approach: almost perfect extraction

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x})\,|\,\mathbf{x}\rangle$





If the i^{th} entry of $\mathbf{x}_1$ is g,
$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c\cdot g)$
$\approx (\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$

Otherwise,
$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c\cdot g)$
$\approx (\mathbf{u}, r)$

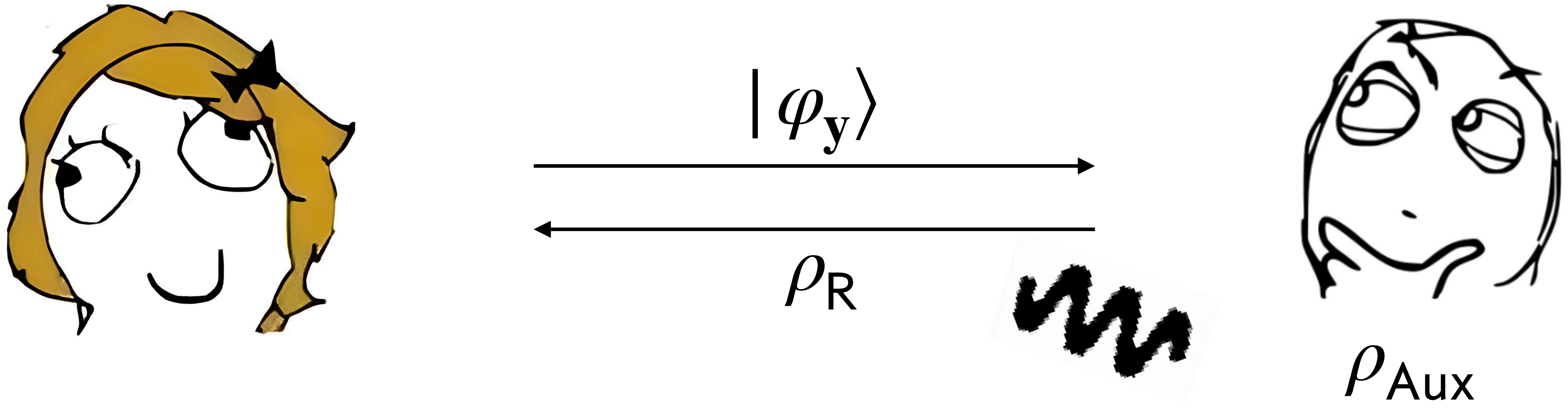
- Test if it is a good distinguisher for $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$ via ATI
- If yes, test it on $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c\cdot g)$ vs $(\mathbf{u}, r)$ for each guess g
  - If the i^{th} entry of $\mathbf{x}_1$ is g, it's a good distinguisher with certainty
  - Otherwise, it's a bad distinguisher with certainty

$\mathbf{i}$ is the unit vector where the i^{th} entry is 1.

Our approach: almost perfect extraction

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x}\in\mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x})|\mathbf{x}\rangle$



If the i^{th} entry of $\mathbf{x}_1$ is g,
$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c \cdot g)$
$\approx (\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$

Otherwise,
$(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c \cdot g)$
$\approx (\mathbf{u}, r)$

As long as the first test passes, the extraction works with certainty.

- Test if it is a good distinguisher for $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$ via ATI
- If yes, test it on $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c \cdot g)$ vs $(\mathbf{u}, r)$ for each guess g
  - If the i^{th} entry of $\mathbf{x}_1$ is g, it's a good distinguisher with certainty
  - Otherwise, it's a bad distinguisher with certainty

$\mathbf{i}$ is the unit vector where the i^{th} entry is 1.

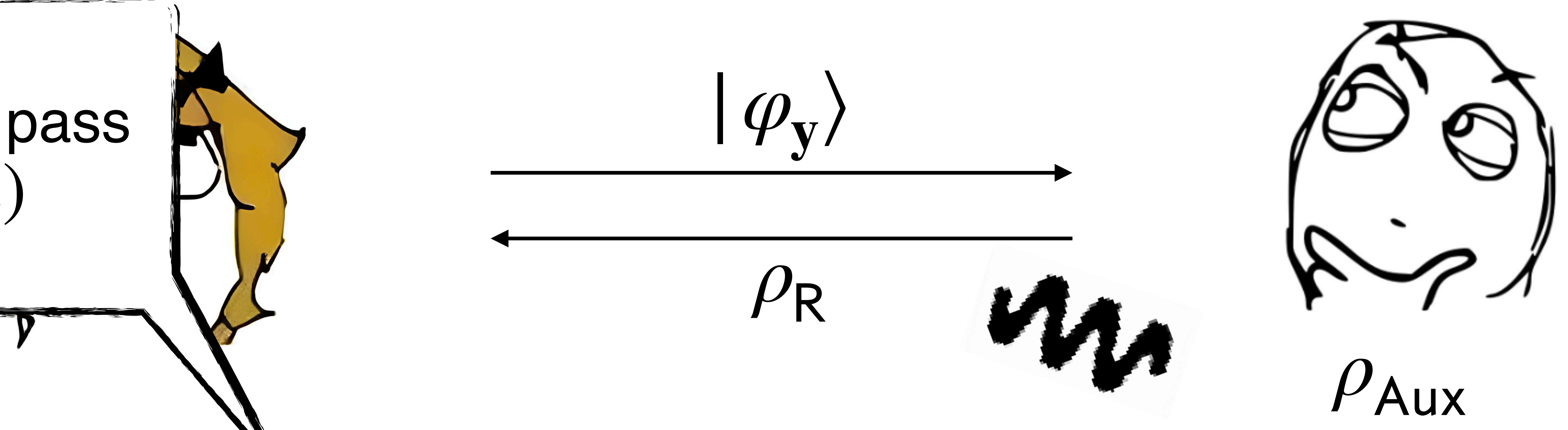
Our approach: almost perfect extraction

The public key is $(\mathbf{A}, \mathbf{y})$ for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and some $\mathbf{y} \in \mathbb{Z}_q^n$

The decryption key is $|\varphi_{\mathbf{y}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{A}\mathbf{x}=\mathbf{y}} \rho_\sigma(\mathbf{x}) \,|\, \mathbf{x}\rangle$

We can get $\mathbf{x}_0$ and $\mathbf{x}_1$ simultaneously!

The test and the revocation pass simultaneously w.p. $1/\text{poly}(n)$



As long as the first test passes, the extraction works with certainty.

- Test if it is a good distinguisher for $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{s}^T\mathbf{y} + e')$ vs $(\mathbf{u}, r)$ via ATI
- If yes, test it on $(\mathbf{s}^T\mathbf{A} + \mathbf{e}^T + c\mathbf{i}, \mathbf{s}^T\mathbf{y} + e' + c \cdot g)$ vs $(\mathbf{u}, r)$ for each guess g
  - If the i^{th} entry of $\mathbf{x}_1$ is g, it's a good distinguisher with certainty
  - Otherwise, it's a bad distinguisher with certainty

$\mathbf{i}$ is the unit vector where the i^{th} entry is 1.

Conclusion

Assuming post-quantum polynomial hardness of LWE over sub-exponential modulus,

- The dual-Regev PKE scheme (the construction in [APV23]) is revocable

+ the results in [APV23]

Assuming post-quantum polynomial hardness of LWE over sub-exponential modulus,

- The dual-Regev PKE scheme has classical revocation
- There exists revocable FHE with quantum/classical revocation
- There exists revocable PRF with quantum/classical revocation

Thank you!

eprint 2024/738