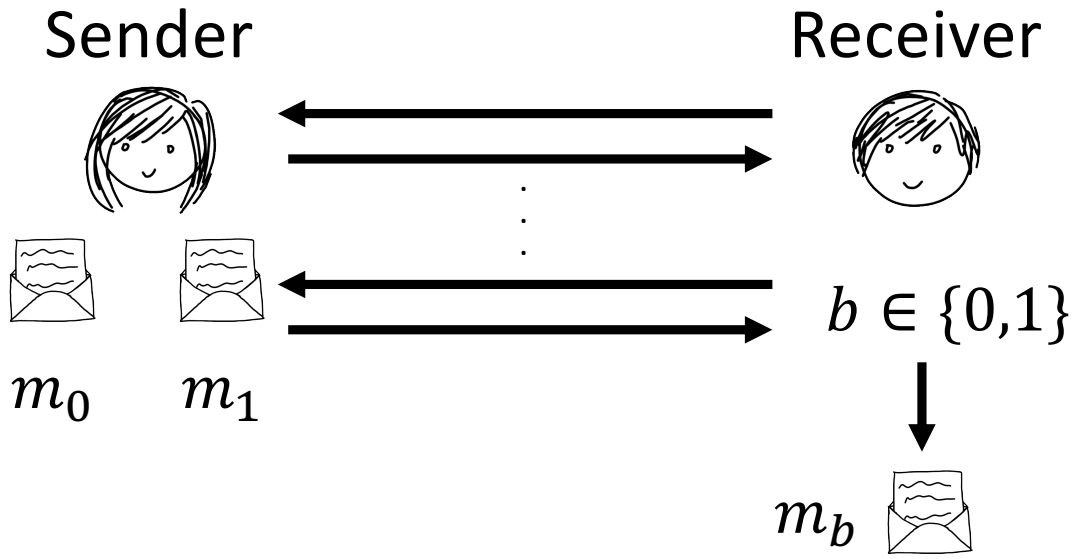
UC-Secure OT from LWE, Revisited

Willy Quach
Northeastern University

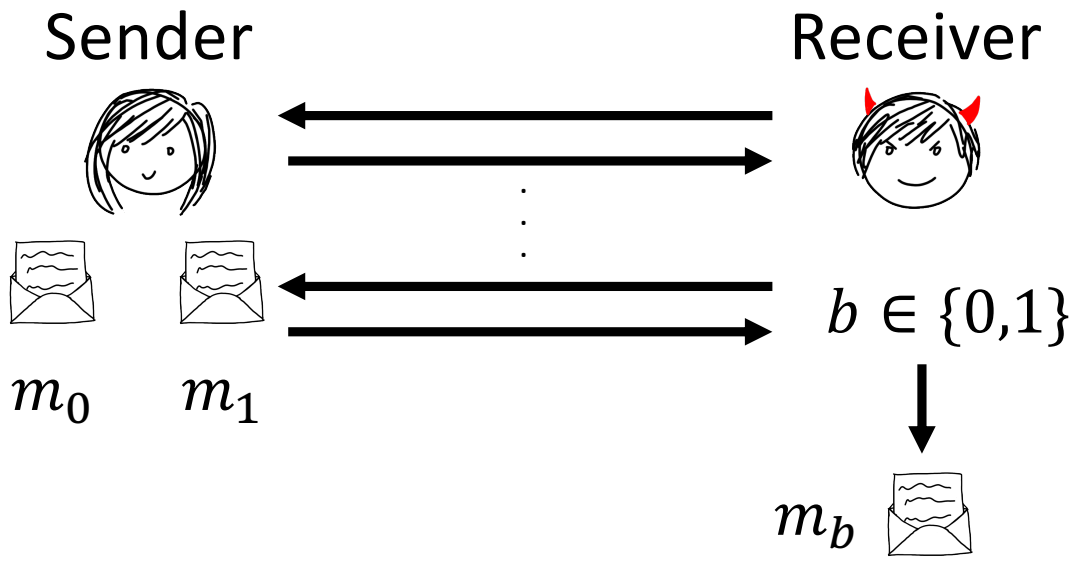
SCN 2020

Oblivious Transfer



Oblivious Transfer

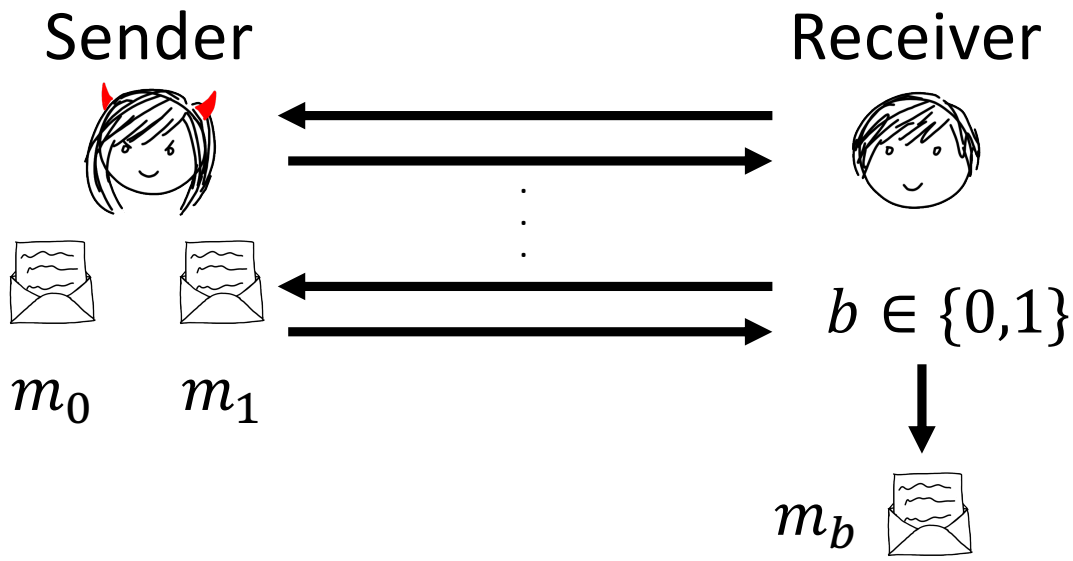
Sender Security: m_{1-b} is hidden



Oblivious Transfer

Sender Security: m_{1-b} is hidden

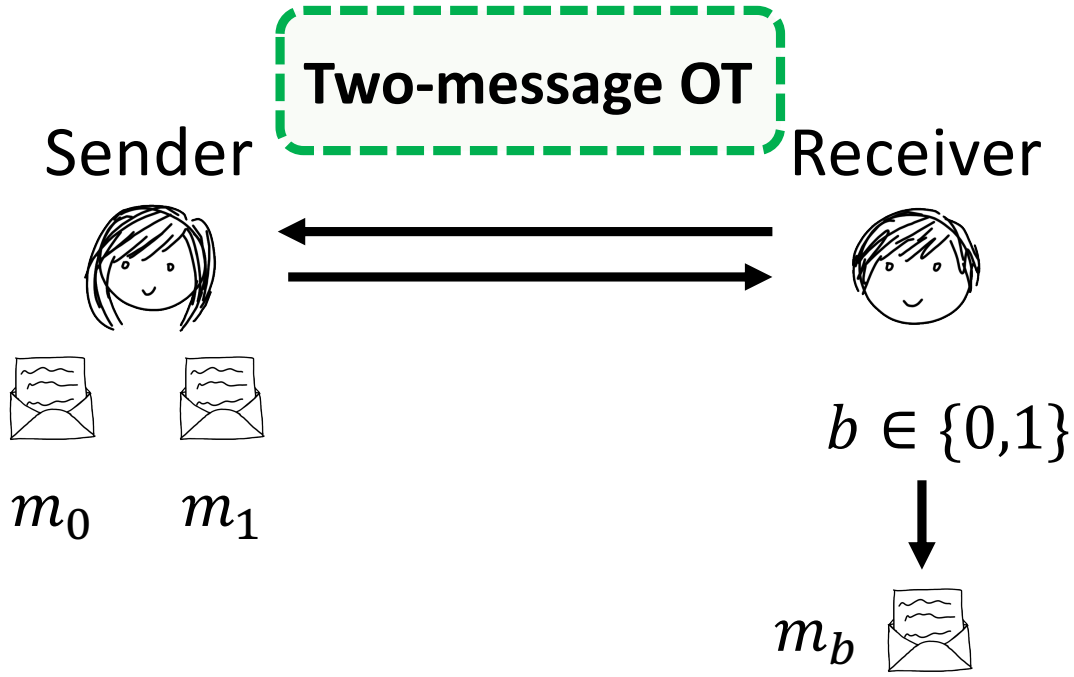
Receiver Security: b is hidden



Oblivious Transfer

Sender Security: m_{1-b} is hidden

Receiver Security: b is hidden

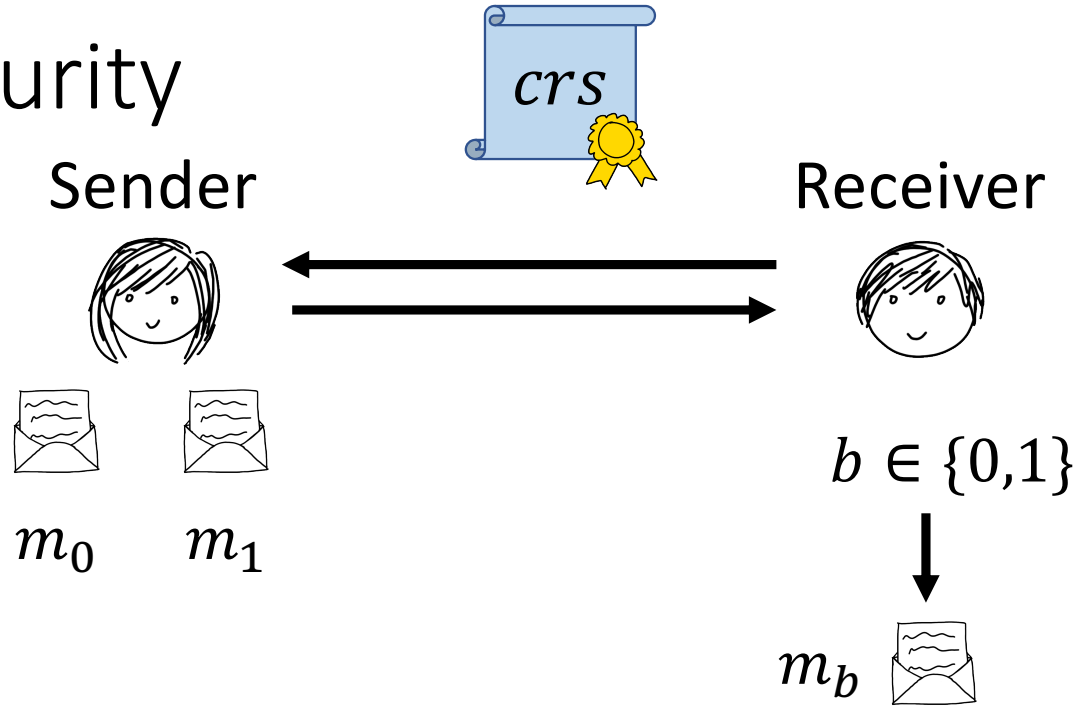


- Fundamental building-block: « MPC-completeness » [Yao86, GMW87]
- Also used in practice!

Different Flavors of Security

Sender Security: m_{1-b} is hidden

Receiver Security: b is hidden

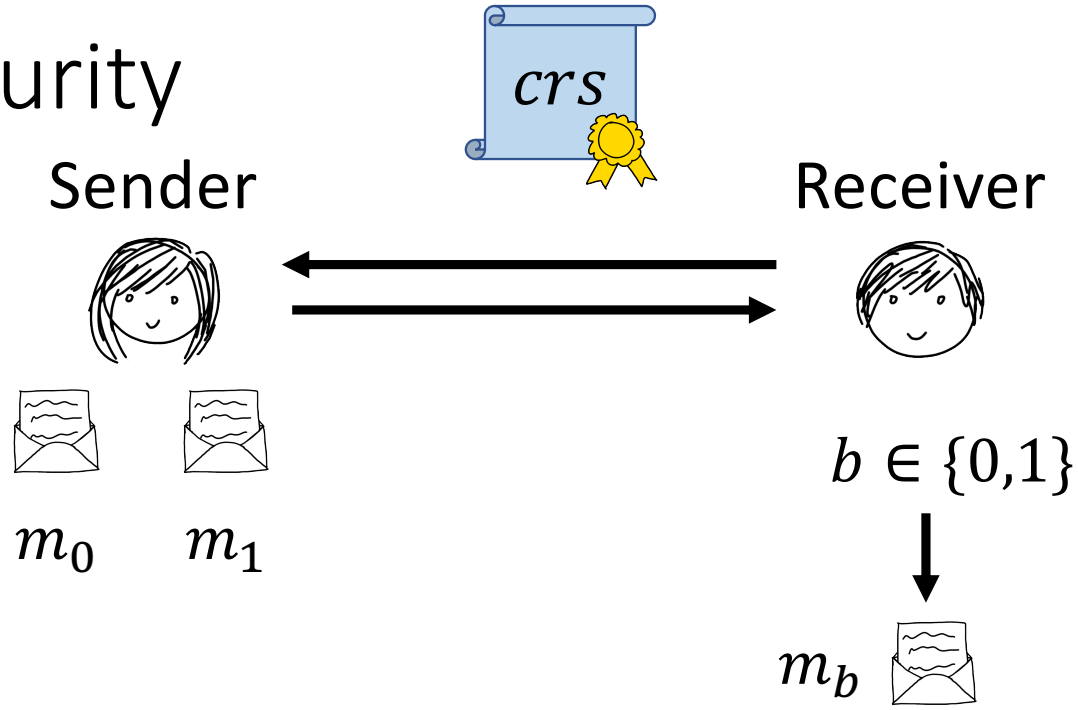


- ~~Semi-honest~~ VS Malicious Adv.
- Computational VS Statistical
  - Statistical for at most one party
- Game-based VS Simulation-based (UC)
  - ~~Not composable Plain model~~
  - Composable Requires Setup

Different Flavors of Security

Sender Security: m_{1-b} is hidden

Receiver Security: b is hidden



- ~~Semi-honest~~ VS **Malicious Adv.**
- Computational VS **Statistical**
  - Statistical for at most one party
- Game-based VS **Simulation-based**
  - ~~Not composable Plain model~~
  - Composable Requires Setup

Prior Work on Malicious 2-Message OT

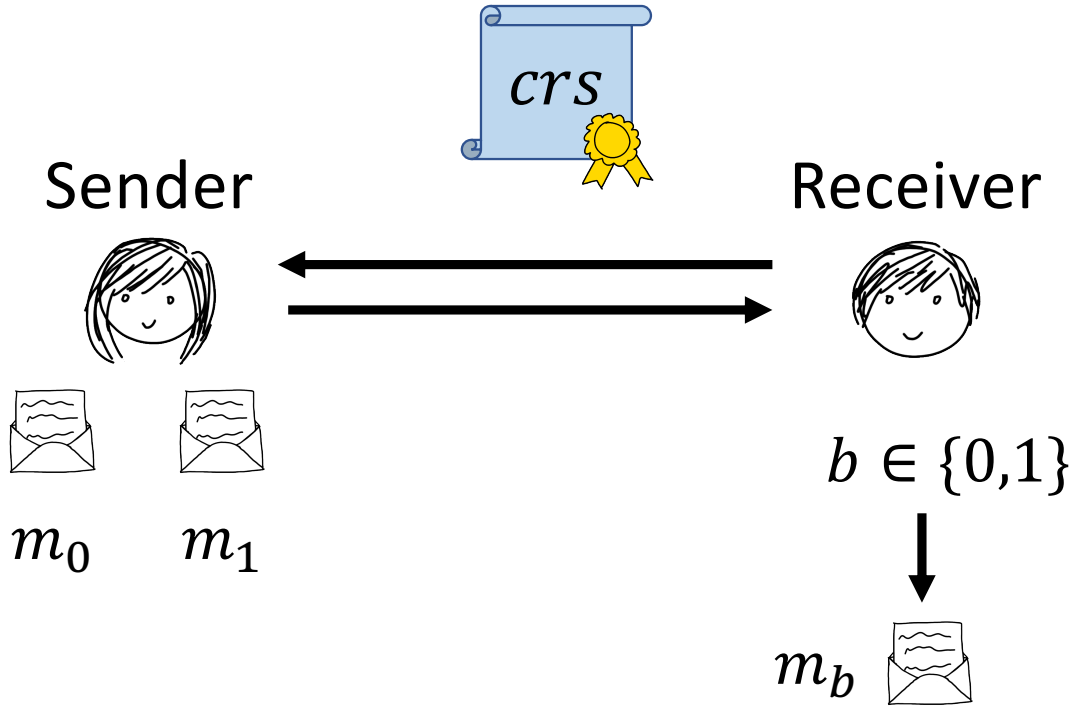
	Ind. VS Simulation	Comp. VS Statistical	Assumption
[Naor-Pinkas01, Aiello-Ishai-Reingold01, Halevi-Kalai12]	Game-based (plain model, Sim. sender)	Comp.	DDH/QR
[Brakerski-Döttling18]	Game-based (plain model)	Statistical sender	**LWE**
[Peikert-Vaikuntanathan-Waters08]	Simulation-based	Either sender or receiver	DDH/QR/**LWE**
[Lindell08]	Simulation-based	Comp.	DDH/QR
[Döttling-Garg-Hajiabadi-Masny-Wichs20]:	Simulation-based	Comp.	CDH/LPN/**LWE**

Prior Work on Malicious 2-Message OT

	Ind. VS Simulation	Comp. VS Statistical	Assumption
[Naor-Pinkas01, Aiello-Ishai-Reingold01, Halevi-Kalai12]	Game-based (plain model, Sim. sender)	Comp.	DDH/QR
[Brakerski-Döttling18]	Game-based (plain model)	Statistical sender	**LWE**
[Peikert-Vaikuntanathan-Waters08]	Simulation-based	Either sender or receiver	DDH/QR/**LWE**
[Lindell08]	Simulation-based	Comp.	DDH/QR
[Döttling-Garg-Hajiabadi-Masny-Wichs20]:	Simulation-based	Comp.	CDH/LPN/**LWE**

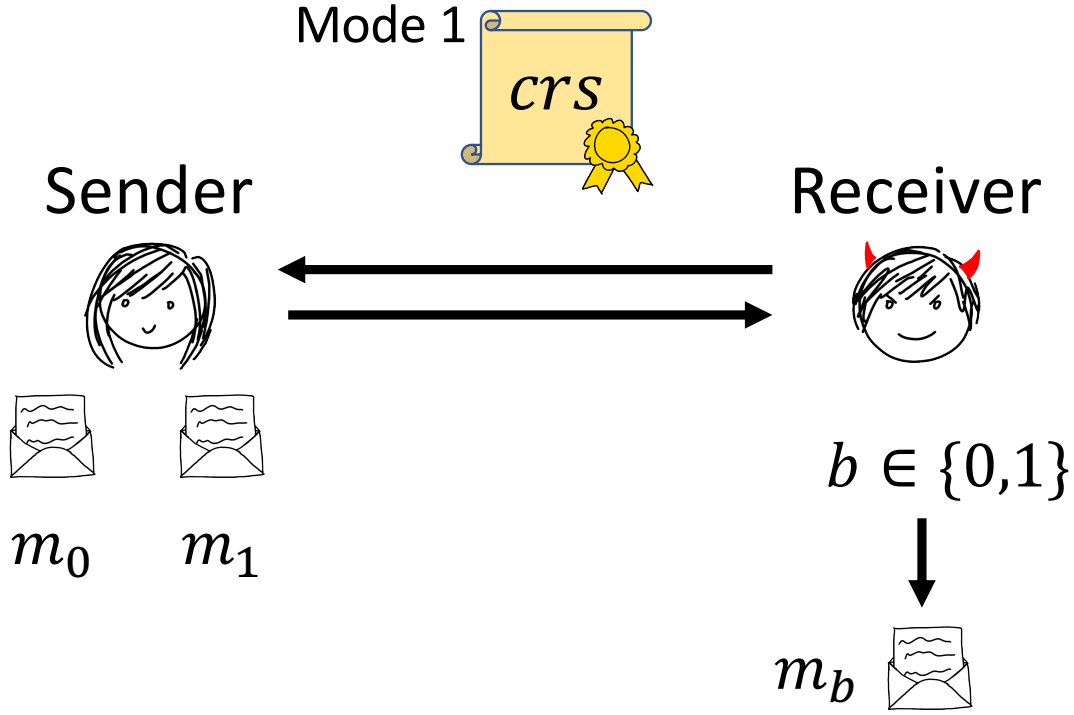
The OT of [PVW08]

- Simulation-secure
- «Dual-Mode» Statistical security
  - Can choose what party gets statistical security at setup!



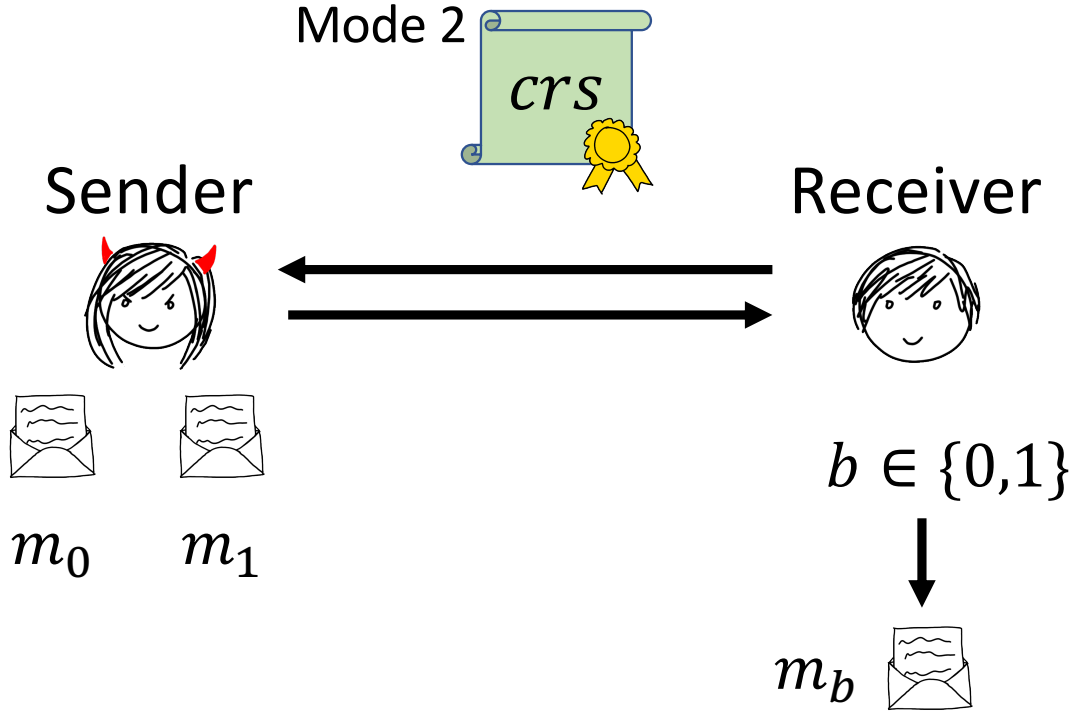
The OT of [PVW08]

- Simulation-secure
- «Dual-Mode» Statistical security
  - Can choose what party gets statistical security at setup!



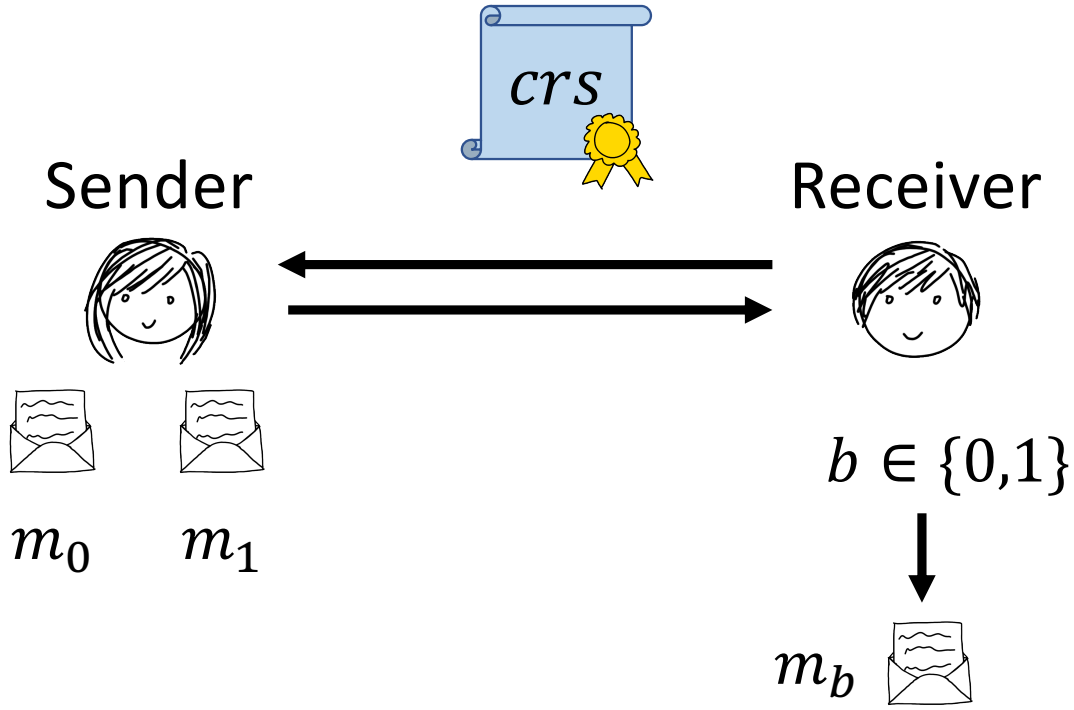
The OT of [PVW08]

- Simulation-secure
- «Dual-Mode» Statistical security
  - Can choose what party gets statistical security at setup!

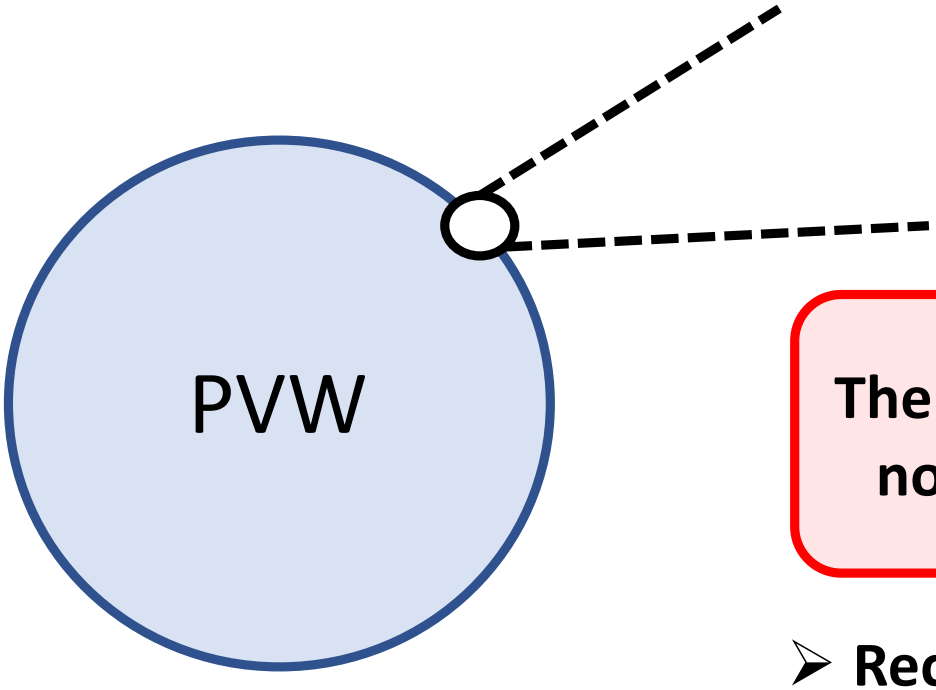


The OT of [PVW08]

- Simulation-secure
- «Dual-Mode» Statistical security
  - Can choose what party gets statistical security at setup!
- Very powerful **blueprint**
- Constructions under DDH/QR/**LWE**, even somewhat efficient!



The OT of PVW from LWE: Abstract View



LWE

The construction from LWE does not fully realize the blueprint

- **Receiver security** only **computational**
- **CRS** can **only be used once**
- No such problems for DDH/QR…

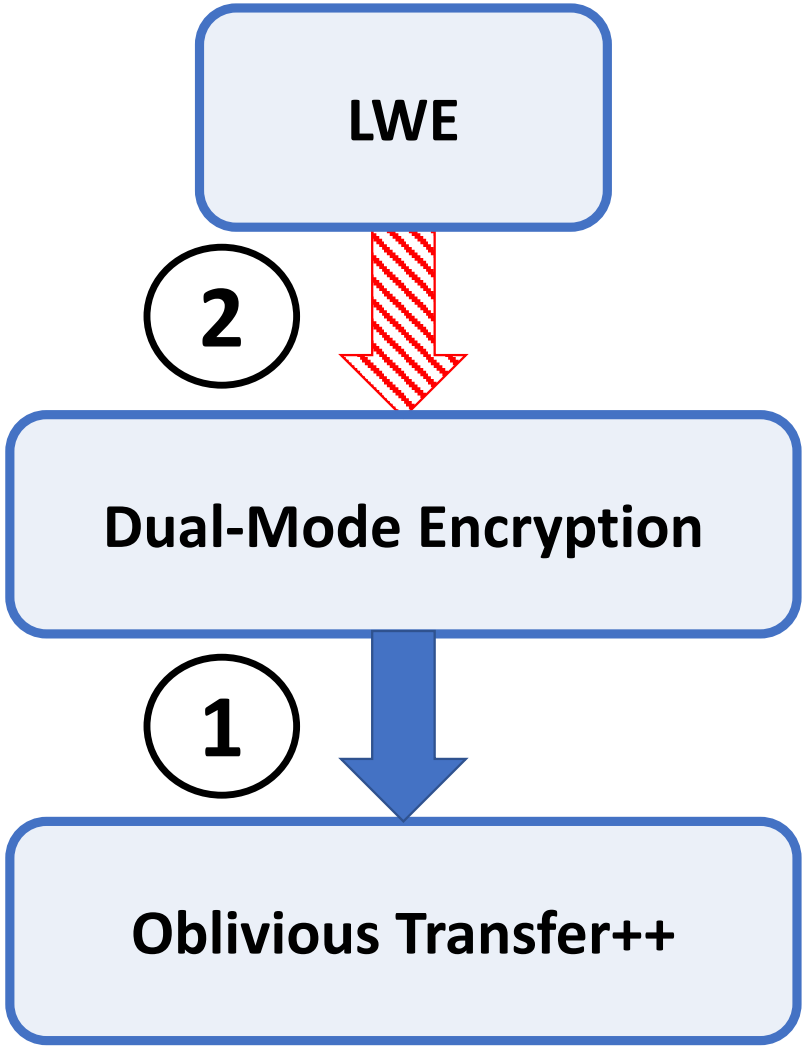
Our Contribution

- **Patch the PVW OT from LWE:**

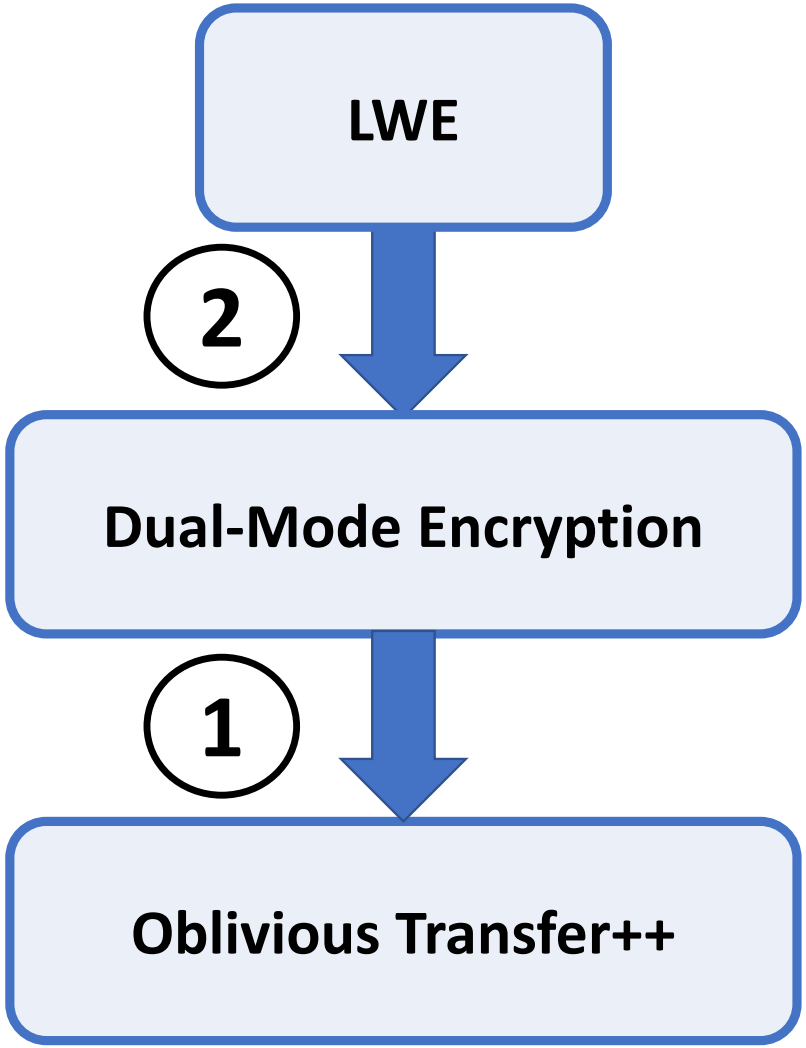
> **Theorem:** There exists a full « dual-mode » simulation-secure OT, assuming LWE with sub-exponential modulus to noise-ratio

- OT with **statistical receiver security** (if CRS is set-up in the appropriate mode)
- CRS is **reusable**
- Under **sub-exponential modulus-to-noise ratio** (PVW: polynomial)

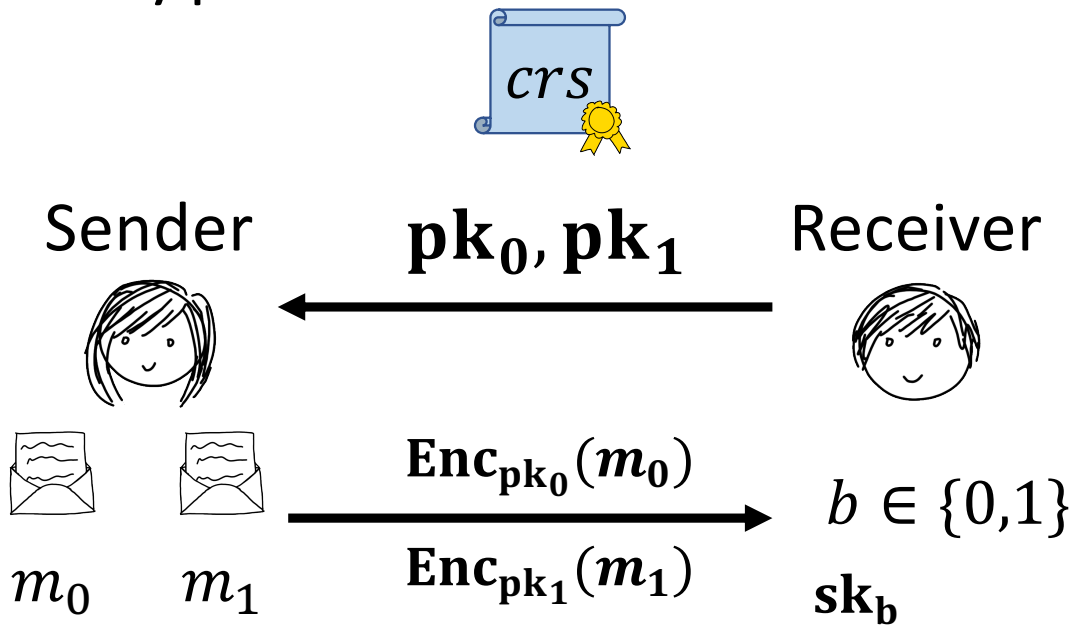
Roadmap



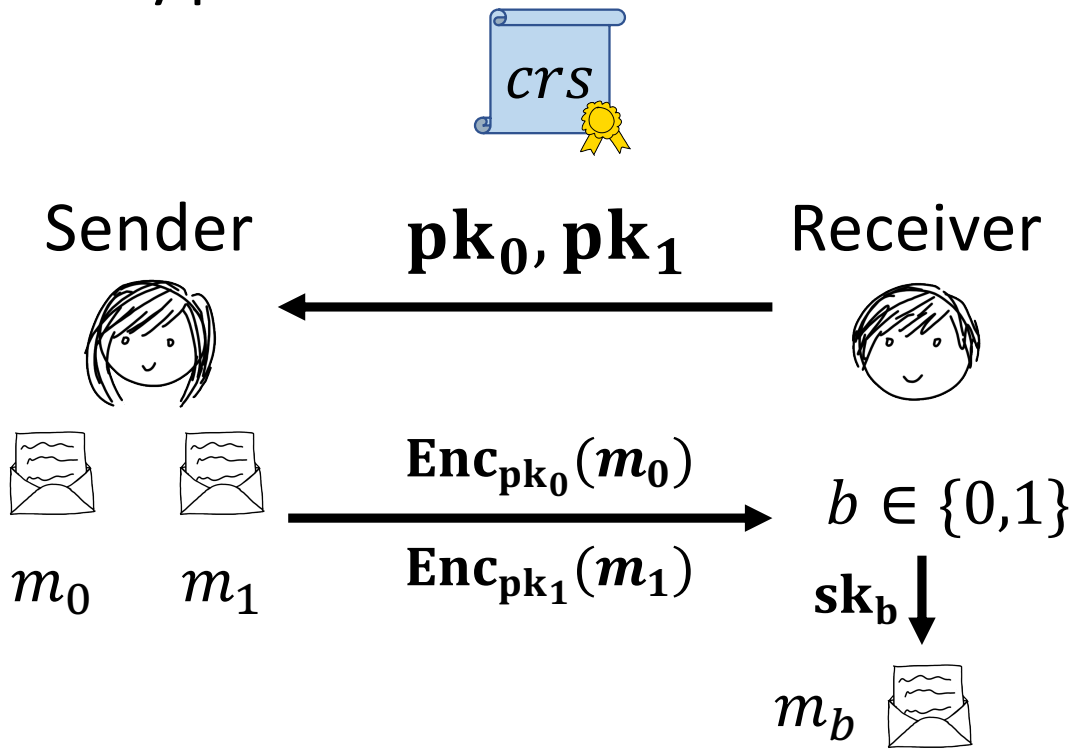
Roadmap



Blueprint: Dual-Mode Encryption

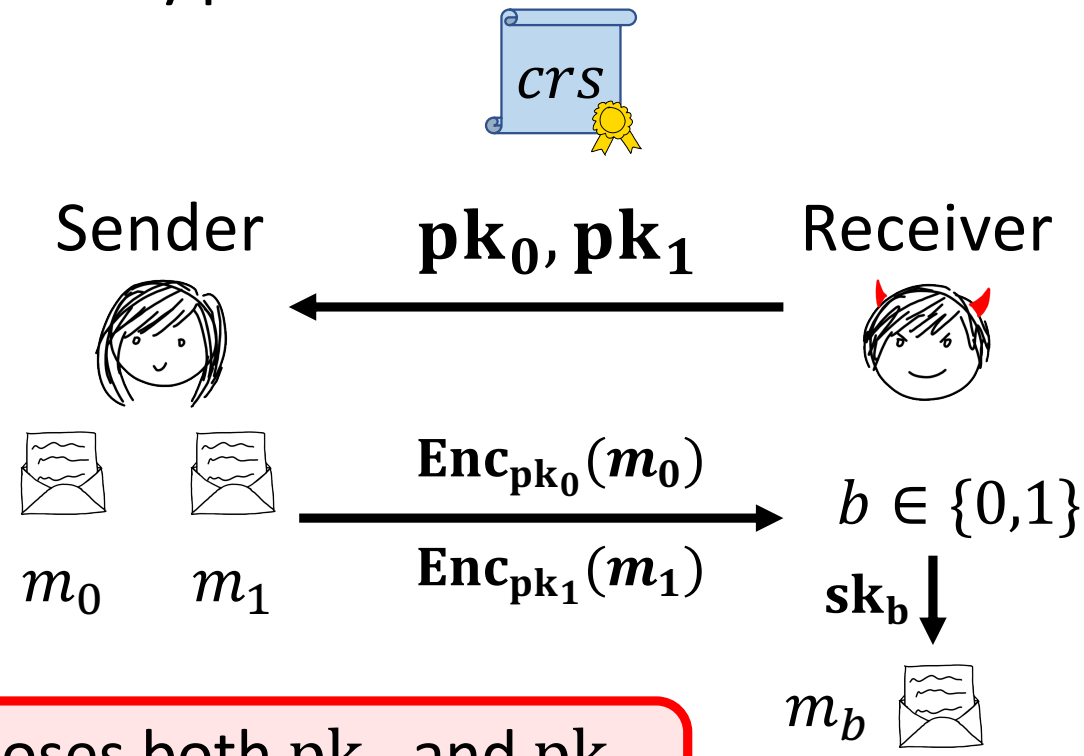


Blueprint: Dual-Mode Encryption



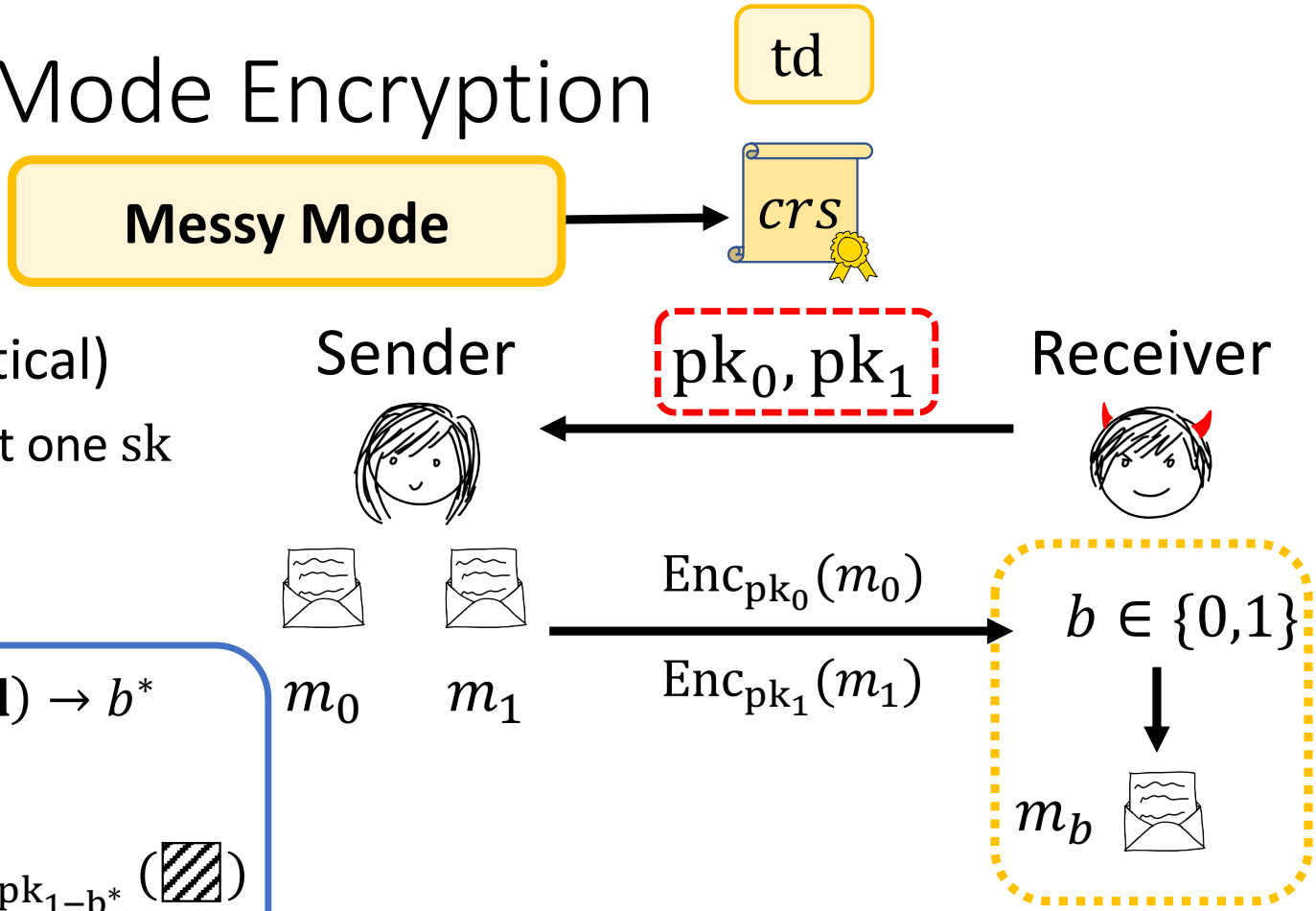
Blueprint: Dual-Mode Encryption

- Main idea: **tie** the public keys together using the CRS



Attack: Receiver chooses both pk_0 and pk_1 with secret keys

Blueprint: Dual-Mode Encryption



- **Sender security**: (statistical)

Receiver should know at most one sk

Simulation security:

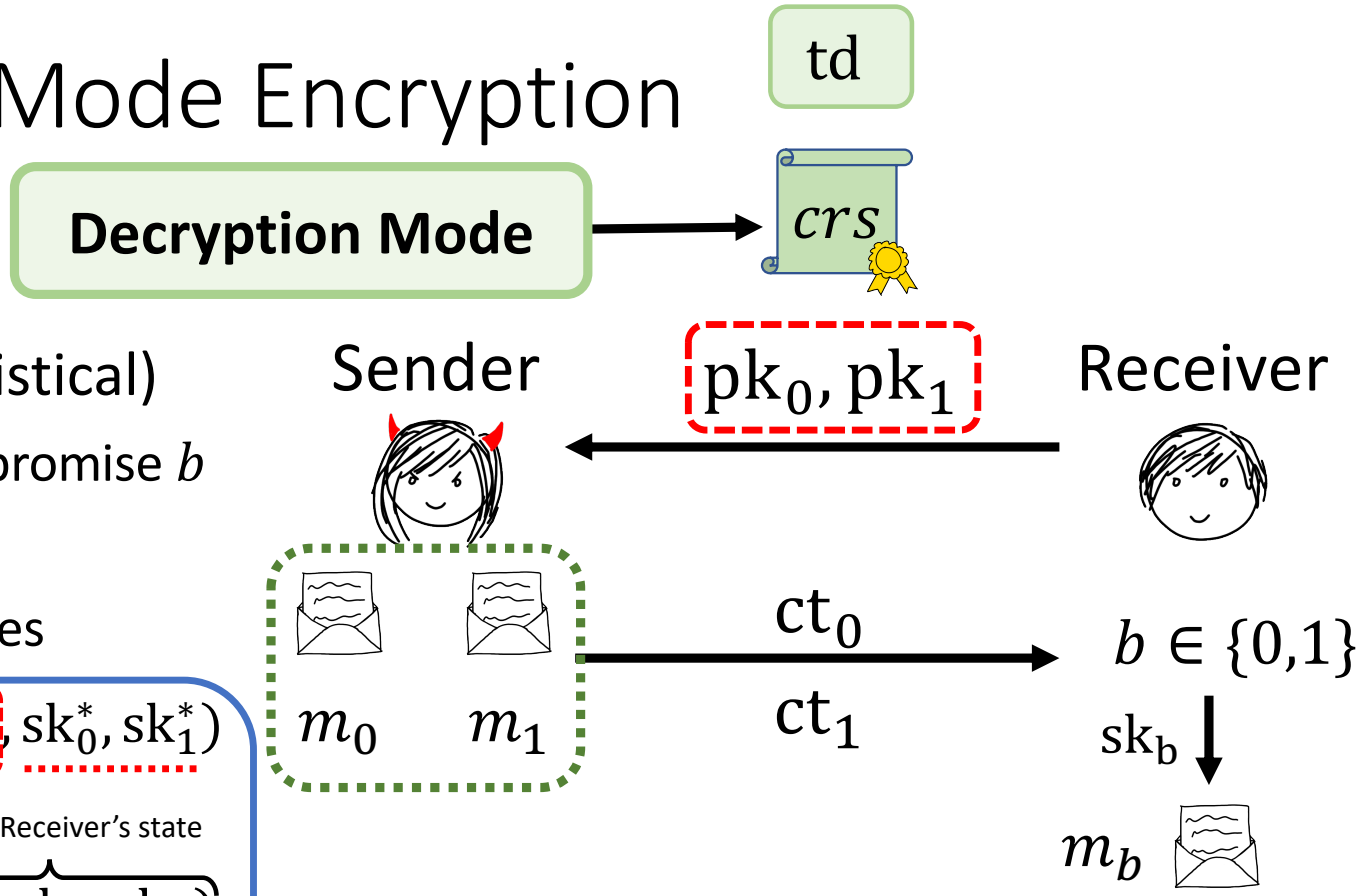
Want to know **which key**:

$$\exists \text{ PPT } \mathrm{Ext}(\mathrm{pk}_0, \mathrm{pk}_1, \mathbf{td}) \to b^*$$

s.t.

$$\mathrm{Enc}_{\mathrm{pk}_{1-b^*}}(m_{1-b^*}) \approx_s \mathrm{Enc}_{\mathrm{pk}_{1-b^*}}(\blacksquare)$$

Blueprint: Dual-Mode Encryption



- **Receiver security**: (statistical)

First message shouldn't compromise b

Simulation security:

Want to **extract** both messages

$\exists$ PPT $\mathrm{Sim}(\mathbf{td}) \to (\mathrm{pk}_0^*, \mathrm{pk}_1^*, \mathrm{sk}_0^*, \mathrm{sk}_1^*)$

s.t.

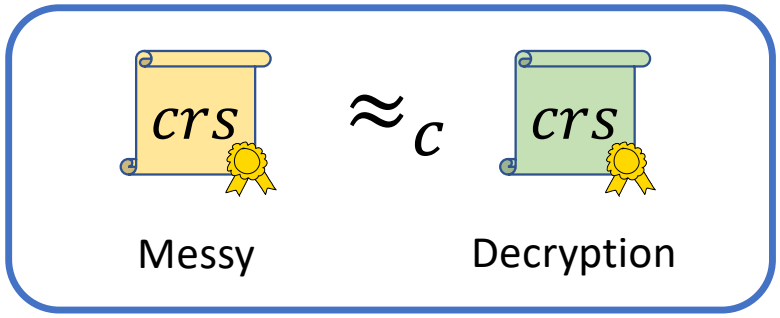
Honest Receiver's state

$$\begin{cases} (\mathrm{pk}_0^*, \mathrm{pk}_1^*, \mathrm{sk}_0^*) \approx_s \overbrace{(\mathrm{pk}_0, \mathrm{pk}_1, \mathrm{sk}_0)} \\ (\mathrm{pk}_0^*, \mathrm{pk}_1^*, \mathrm{sk}_1^*) \approx_s (\mathrm{pk}_0, \mathrm{pk}_1, \mathrm{sk}_1) \end{cases}$$

Blueprint: Dual-Mode Encryption

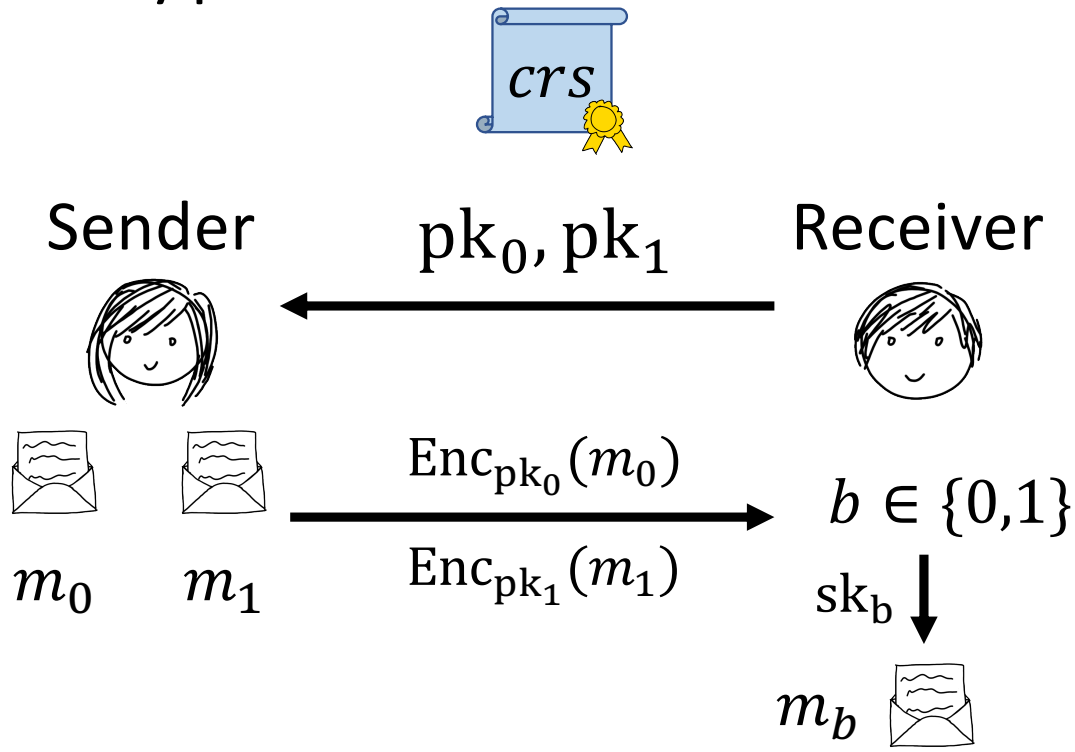
- **Mode indistinguishability:**

The CRS in the two modes are computationally indistinguishable:



⇒ (comp.) receiver security in messy mode

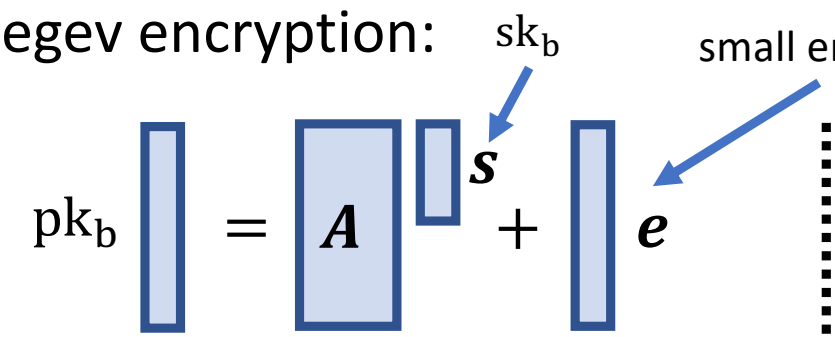
⇒ (comp.) sender security in decryption mode



Dual-Mode Encryption from LWE (1) [PVW]

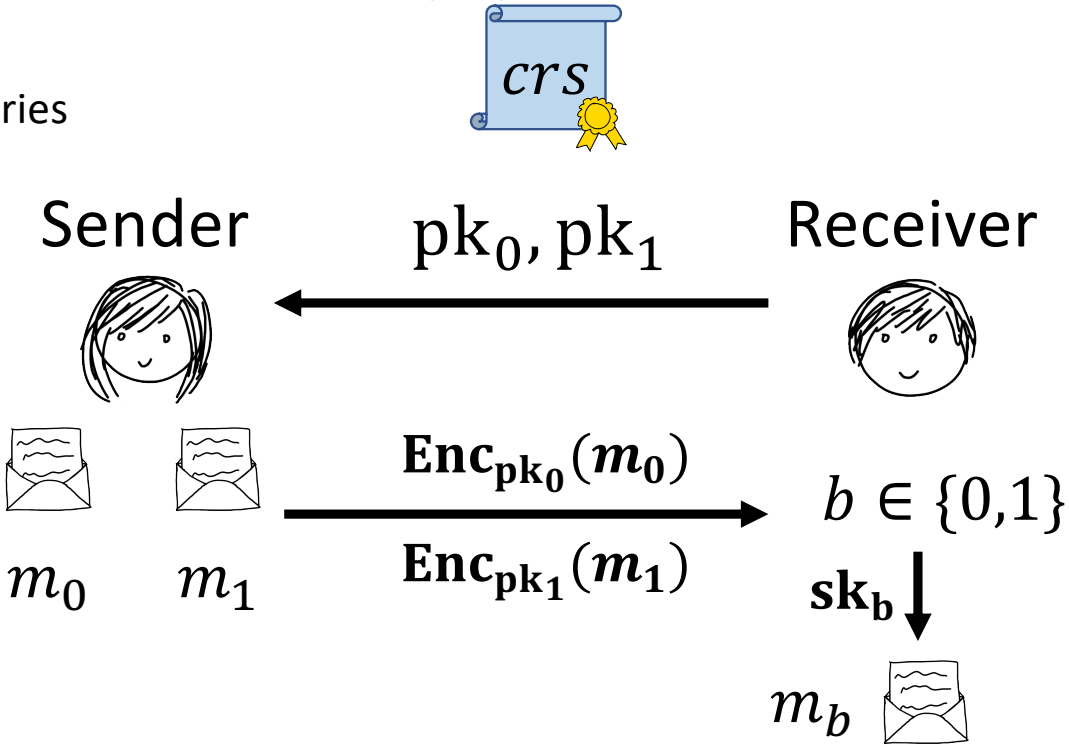
- Regev encryption:

Over $\mathbb{Z}_q$:



small entries

$$\mathrm{Enc}_{\mathrm{pk_b}}(m) = \boldsymbol{r}\,\boldsymbol{A} + \boldsymbol{r}\,\mathrm{pk_b} + \frac{q}{2}\cdot m$$



$$\mathbf{pk_1} = \mathbf{pk_0} + \boldsymbol{crs}$$

Dual-Mode Encryption from LWE (2) [PVW]

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $\mathrm{Enc}_{\mathrm{pk_b}}(m) = (\boldsymbol{rA}, \boldsymbol{rc} + \left(\frac{q}{2}\right) m)$

- **Sender security:**

  Obs: For most $\boldsymbol{c} \in \mathbb{Z}_q^n$:

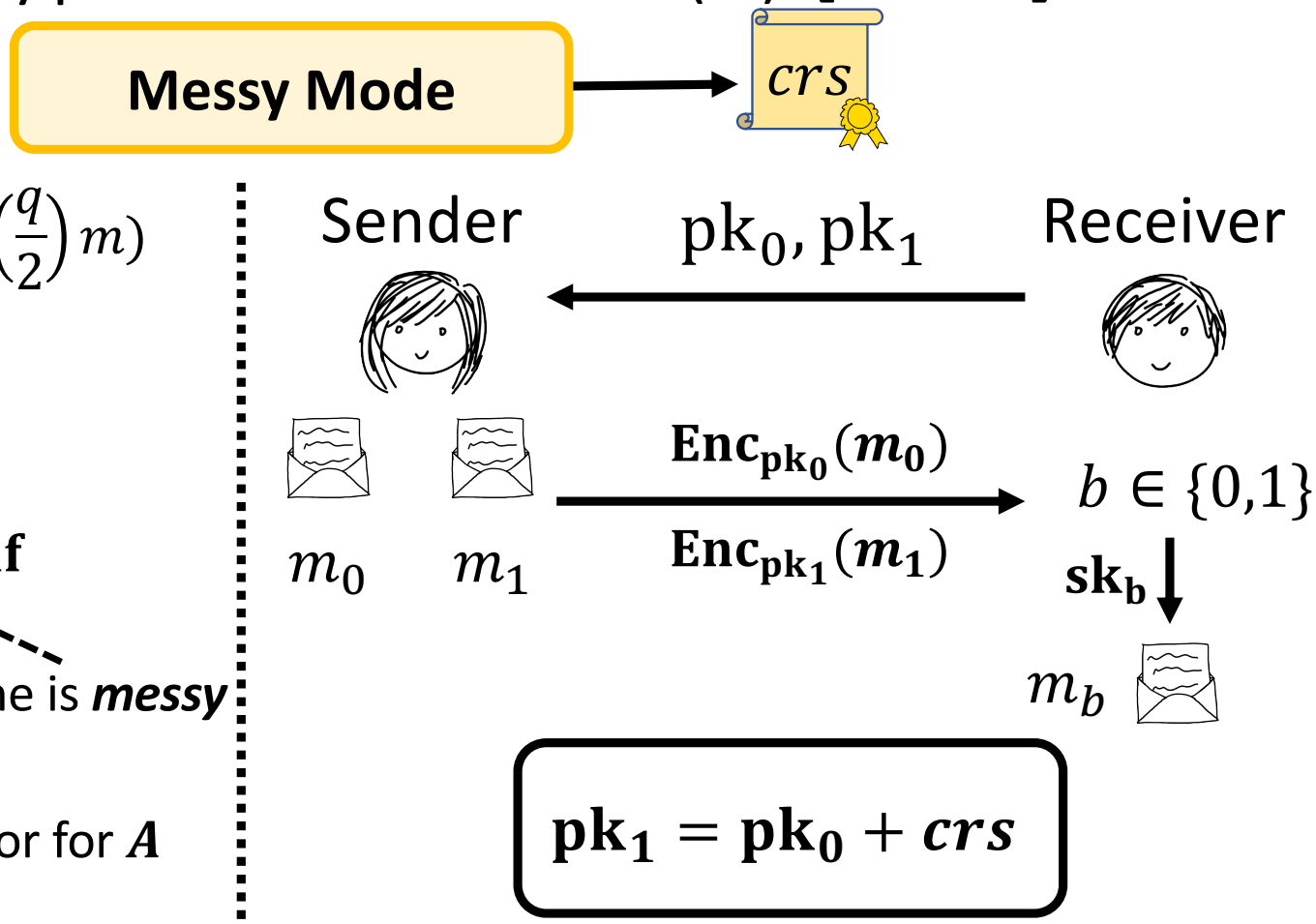
  $$\mathbf{Enc_c}(\boldsymbol{m}) \approx_s \mathbf{unif}$$

  ➢ Set uniform crs!

  ⇒ For **all** pk_0 (at least) one is ***messy***

Can **test** such pk with trapdoor for $\boldsymbol{A}$

[GPV08, MP12]



Dual-Mode Encryption from LWE (2) [PVW]

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $\mathrm{Enc}_{\mathrm{pk_b}}(m) = (\boldsymbol{rA}, \boldsymbol{rc} + \left(\frac{q}{2}\right) m)$

- **Receiver security:**

  Want $\mathrm{pk}_0^*, \mathrm{pk}_1^*$ with **both** secret keys

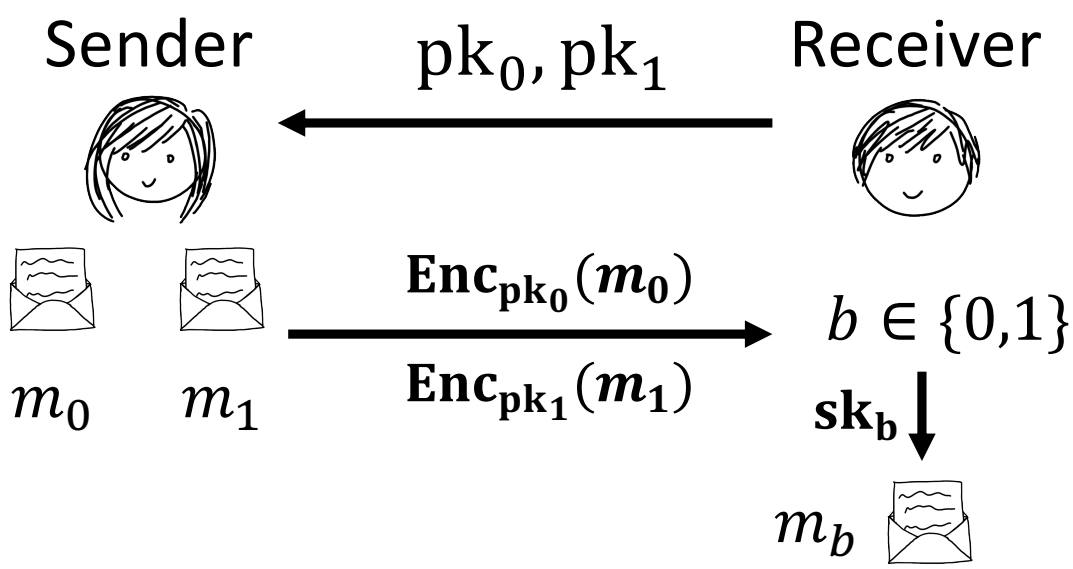
  ➢ Set crs as the difference!

  ⇒ Mode indist. by LWE

Problem 1: only one pair $(\mathbf{pk}_0^*, \mathbf{pk}_1^*)$

Problem 2: keys are not perfectly fresh

Decryption Mode → crs



$$\boldsymbol{crs} = \mathbf{pk}_1^* - \mathbf{pk}_0^*$$

Dual-Mode Encryption from LWE (2) [PVW]

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $crs = (\boldsymbol{A s_1^*} + \boldsymbol{e_1^*}) - (\boldsymbol{A s_0^*} + \boldsymbol{e_0^*})$

  $= \boldsymbol{A}(\boldsymbol{s_1^*} - \boldsymbol{s_0^*}) + (\boldsymbol{e_1^*} - \boldsymbol{e_0^*})$

- **Receiver security:**

  Want $\mathrm{pk}_0, \mathrm{pk}_1$ with **both** secret keys

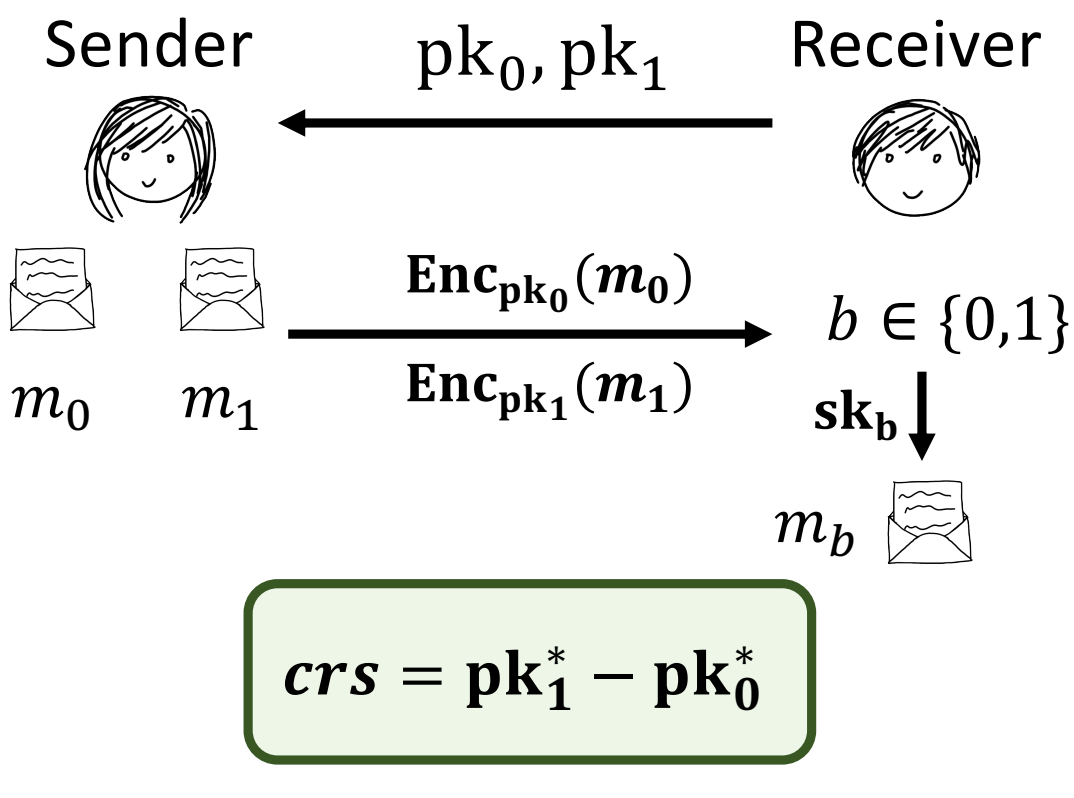
  ➢ Set crs as the difference!

  ⇒ Mode indist. by LWE

Problem 1: only one pair $(\mathbf{pk_0^*}, \mathbf{pk_1^*})$

Problem 2: keys are not perfectly fresh

Decryption Mode → crs



$$\boldsymbol{crs} = \mathbf{pk_1^*} - \mathbf{pk_0^*}$$

Dual-Mode Encryption from LWE (2) [PVW]

- Regev encryption

$$\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$$

$$crs = (\boldsymbol{A s_1^*} + \boldsymbol{e_1^*}) - (\boldsymbol{A s_0^*} + \boldsymbol{e_0^*})$$

$$= \boldsymbol{A}(\boldsymbol{s_1^*} - \boldsymbol{s_0^*}) + (\boldsymbol{e_1^*} - \boldsymbol{e_0^*})$$

- **Receiver security:**

  Want $\mathrm{pk}_0, \mathrm{pk}_1$ with **both** secret keys

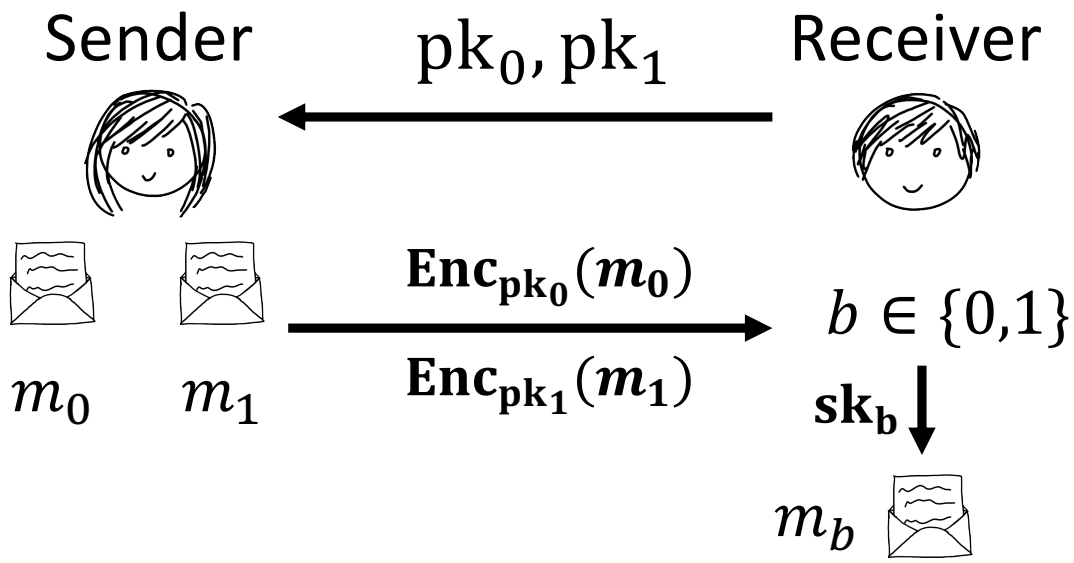
  ➢ Set crs as the difference!

  ⇒ Mode indist. by LWE

Problem 1: CRS is not reusable

Problem 2: Computational security

Decryption Mode → crs



$$\boldsymbol{crs} = \mathbf{pk_1^*} - \mathbf{pk_0^*}$$

Fix 1: Noise flooding (~ Standard)

- Regev encryption

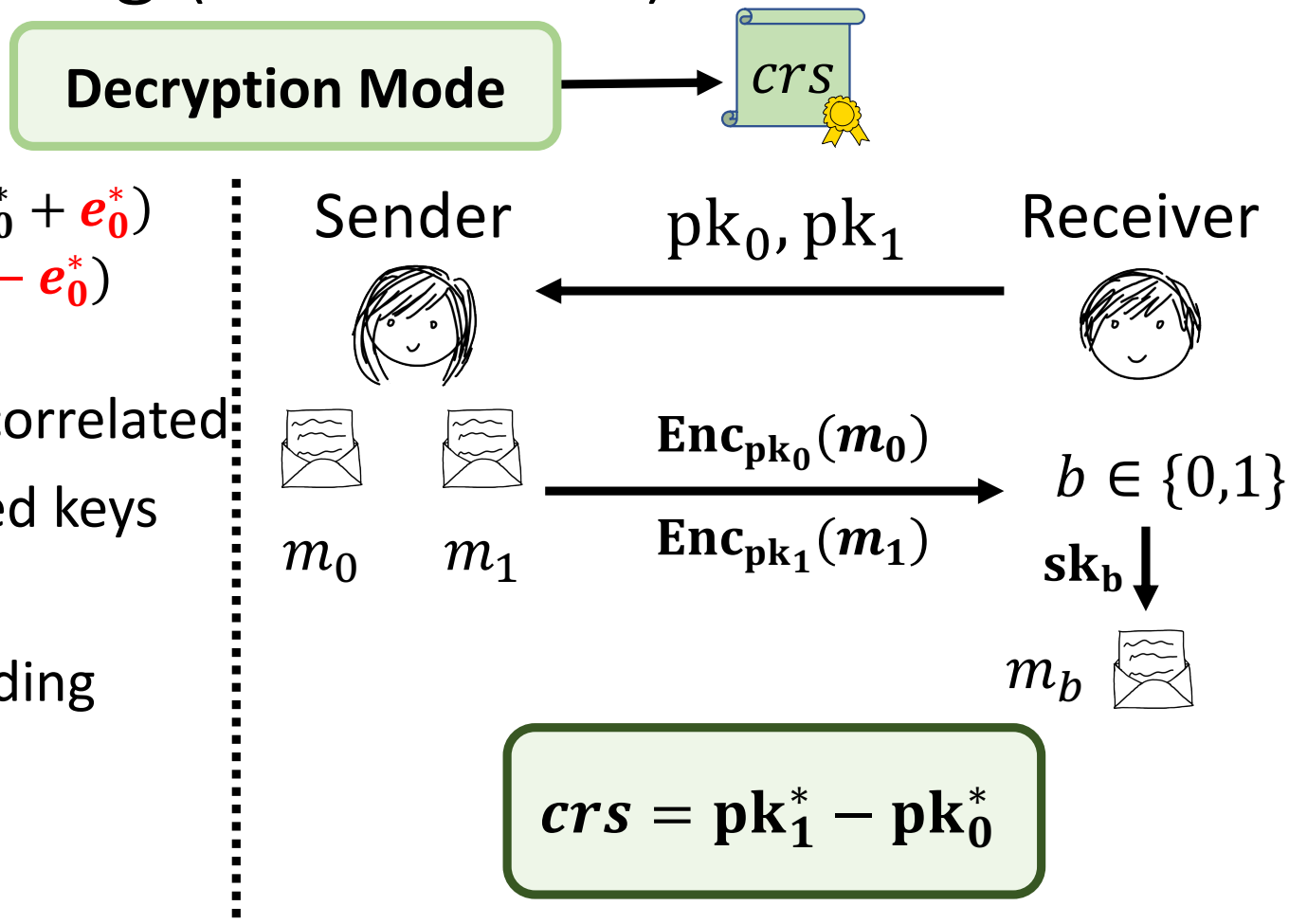
$$\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$$

$$crs = (\boldsymbol{A s_1^*} + \boldsymbol{e_1^*}) - (\boldsymbol{A s_0^*} + \boldsymbol{e_0^*})$$

$$= \boldsymbol{A}(\boldsymbol{s_1^*} - \boldsymbol{s_0^*}) + (\boldsymbol{e_1^*} - \boldsymbol{e_0^*})$$

- **Problem:** noise of crs correlated with noise of trapdoored keys

- Standard fix: noise flooding



Fix 1: Noise flooding (~ Standard)

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $crs = \boldsymbol{A s^*} + \boldsymbol{e^*}$

  $sk^*_{1-b} = \boldsymbol{s_b} + \boldsymbol{s^*}$

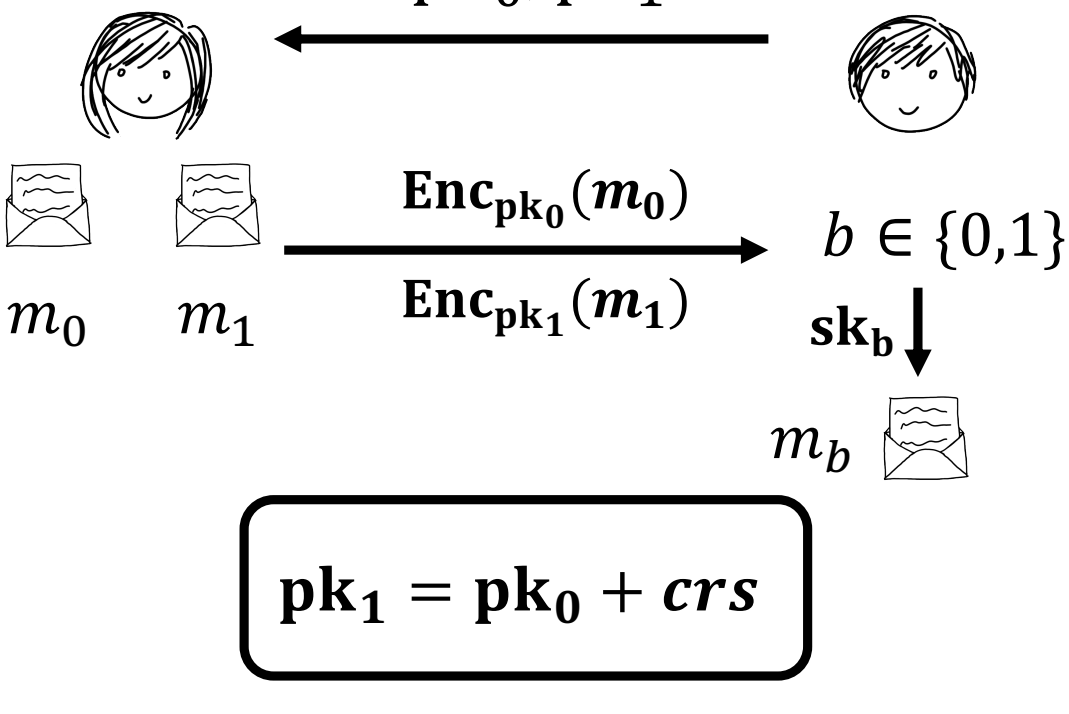
  $\|\boldsymbol{e}\| \gg \|\boldsymbol{e^*}\|$

- ~~**Problem:** noise of crs correlated with noise of trapdoored keys~~

- Standard fix: noise flooding
- Solves both problems!
  - Cost: sub-exponential LWE modulus q

    ⇒ stronger assumption

Decryption Mode → crs



$\mathbf{pk_1} = \mathbf{pk_0} + \boldsymbol{crs}$

Back to Sender Security…

- Regev encryption

$$\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$$

$$\mathrm{Enc}_{\mathrm{pk_b}}(m) = (\boldsymbol{rA}, \boldsymbol{rc} + \left(\frac{q}{2}\right) m)$$

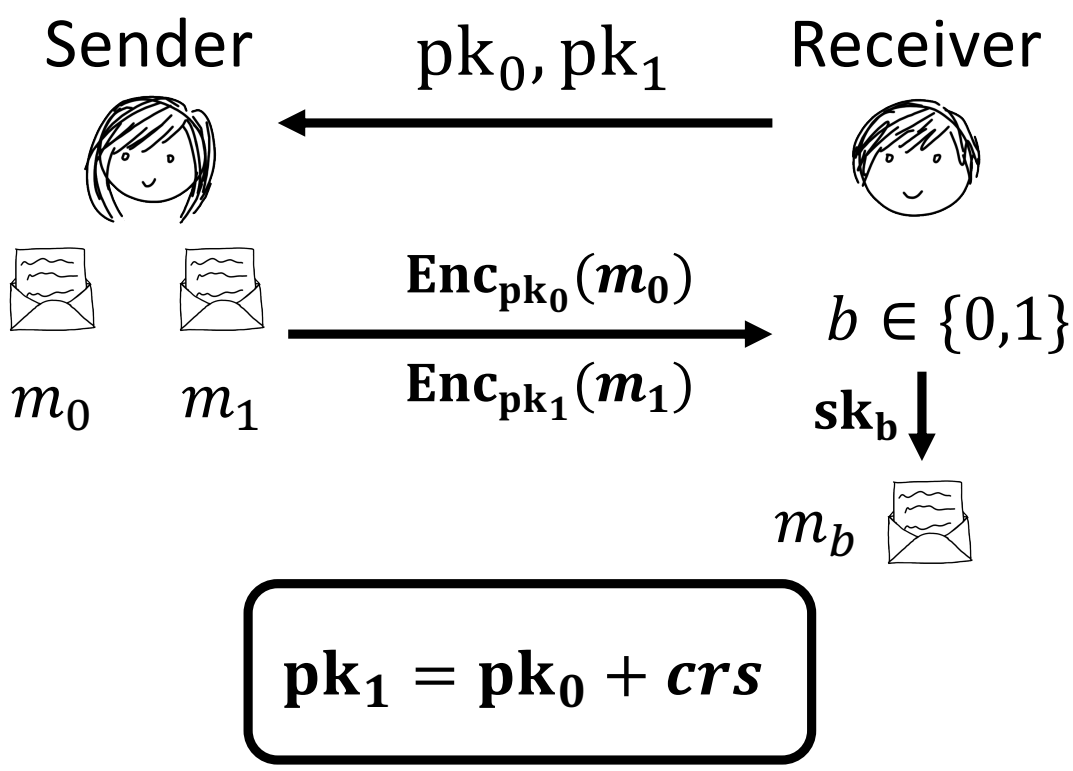
- **Sender Security:**

  Obs: For most $\boldsymbol{c}$, $\mathbf{Enc_c}(\boldsymbol{m}) \approx_s$ **unif**

  Can **test** such $\boldsymbol{c}$ with trapdoor for $\boldsymbol{A}$

New Problem:
This testing time is **linear in the modulus**
⇒ **Extractor now runs in sub-exp. time**

Messy Mode → crs



$$\mathbf{pk_1} = \mathbf{pk_0} + \boldsymbol{crs}$$

Main Idea: Randomized Rounding

Messy Mode

- Regev encryption

$$\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$$

$$\mathrm{Enc}_{\mathrm{pk_b}}(m) = (\boldsymbol{rA}, \boldsymbol{rc} + \left(\frac{q}{2}\right) m)$$

- **Sender Security:**

  Want: efficient « messiness test »

$$\mathbf{pk} \mid \mathbf{Enc_{pk}}(\boldsymbol{m}) \approx \mathbf{Enc_{pk}}(\blacksquare)$$

Don’t know better test for Regev…

Main Idea: Randomized Rounding

Messy Mode

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $\mathrm{Enc}_{\mathrm{pk_b}}(m) =$ $(\boldsymbol{rA}, \boldsymbol{rc} + \left(\frac{q}{2}\right) m)$

- **Sender Security:**

  Want: efficient « messiness test »

  $\mathbf{pk} \mid \mathbf{Enc_{pk}}(\boldsymbol{m}) \approx \mathbf{Enc_{pk}}(\boxed{})$

Don't know better test for Regev…
Can change encryption scheme!
(only used structure on public/secret keys so far)
Goal: **simple** « messiness » property

Claim [Benhamouda-Blazy-Ducas-Q18]

Randomized rounding R s.t.:

1. If $\boldsymbol{c} = \boldsymbol{As} + \boldsymbol{e}$ for some small $\boldsymbol{e}$,

   then $\left(\boldsymbol{rA}, R(\boldsymbol{rc})\right) \simeq (\boldsymbol{rA}, R(\boldsymbol{rAs}))$

Main Idea: Randomized Rounding

Messy Mode

- Regev encryption

$\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

$\mathrm{Enc}_{\mathrm{pk_b}}(m) = (\boldsymbol{rA}, \boldsymbol{rc} + \left(\frac{q}{2}\right) m)$

- **Sender Security:**

Want: efficient « messiness test »

$\mathbf{pk} \mid \mathbf{Enc_{pk}}(\boldsymbol{m}) \approx \mathbf{Enc_{pk}}(\blacksquare)$

Don't know better test for Regev…
Can change encryption scheme!
(only used structure on public/secret keys so far)
Goal: **simple** « messiness » property

Claim [Benhamouda-Blazy-Ducas-Q18]

Randomized rounding R s.t.:

1. If $\boldsymbol{c} = \boldsymbol{As} + \boldsymbol{e}$ for some small $\boldsymbol{e}$,
   then $\left(\boldsymbol{rA}, R(\boldsymbol{rc})\right) \simeq (\boldsymbol{rA}, R(\boldsymbol{rAs}))$

   ⇒ Correctness

2. If $\boldsymbol{c} \neq \boldsymbol{As} + \boldsymbol{e}$ for any large $\boldsymbol{e}$,
   then $\left(\boldsymbol{rA}, R(\boldsymbol{rc})\right) \approx (\boldsymbol{rA}, \mathrm{unif})$

Idea: Randomized Rounding

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $\mathrm{Enc}_{\mathrm{pk_b}}(m) =$ $\boldsymbol{rA}$, $R(\boldsymbol{rc}) \oplus m$

- **Sender Security:**

  Want: efficient « messiness test »

  $\mathbf{pk} \mid \mathbf{Enc_{pk}}(\boldsymbol{m}) \approx \mathbf{Enc_{pk}}(\blacksquare)$

Simpler messiness property:
Can test in time indep. of modulus using a trapdoor [GPV08, MP12]

$\Rightarrow \boldsymbol{c}$ is messy

Messy Mode

Claim [Benhamouda-Blazy-Ducas-Q18]

Randomized rounding R s.t.:

1. If $\boldsymbol{c} = \boldsymbol{As} + \boldsymbol{e}$ for some small $\boldsymbol{e}$,
   then $(\boldsymbol{rA}, R(\boldsymbol{rc})) \simeq (\boldsymbol{rA}, R(\boldsymbol{rAs}))$

$\Rightarrow$ Correctness

2. If $\boldsymbol{c} \neq \boldsymbol{As} + \boldsymbol{e}$ for any large $\boldsymbol{e}$,
   then $(\boldsymbol{rA}, R(\boldsymbol{rc})) \approx (\boldsymbol{rA}, \mathrm{unif})$

Idea: Randomized Rounding

- Regev encryption

  $\mathrm{pk_b} = \boldsymbol{c} \stackrel{\text{def}}{=} \boldsymbol{A s_b} + \boldsymbol{e}$

  $\mathrm{Enc}_{\mathrm{pk_b}}(m) =$ $\boldsymbol{r_i A}$, $R(\boldsymbol{r_i c}) \oplus m$ *

- **Sender Security:**

  Want: efficient « messiness test »

  $\mathbf{pk} \mid \mathbf{Enc_{pk}}(\boldsymbol{m}) \approx \mathbf{Enc_{pk}}(\blacksquare)$

Simpler messiness property:
Can test in time indep. of modulus using a trapdoor [GPV08, MP12]

⇒ $\boldsymbol{c}$ is messy

Messy Mode

Claim [Benhamouda-Blazy-Ducas-Q18]

Randomized rounding R s.t.:

1. If $\boldsymbol{c} = \boldsymbol{A s} + \boldsymbol{e}$ for some small $\boldsymbol{e}$,
   then $(\boldsymbol{rA}, R(\boldsymbol{rc})) \simeq^{*} (\boldsymbol{rA}, R(\boldsymbol{rAs}))$

   ⇒ Correctness

2. If $\boldsymbol{c} \neq \boldsymbol{A s} + \boldsymbol{e}$ for any large $\boldsymbol{e}$,
   then $(\boldsymbol{rA}, R(\boldsymbol{rc})) \approx (\boldsymbol{rA}, \mathrm{unif})$

Summary

- New dual-mode encryption from LWE (sub-exp. modulus-to-noise ratio)

  ⇒ Simulation-secure, « dual-mode » OT from LWE with
  - Statistical security for **either party**, depending on setup
    (First construction that achieves **statistical receiver security**)
  - **Reusable CRS** (only achieved recently from LWE: [DGHMW20])
  - **Sub-exp. modulus-to-noise ratio**
- **Main tool:** Regev encryption with « nice » messiness property
- **Q:** polynomial modulus?
- **Q:** Other applications of the rounding function?

Thank you!

https://eprint.iacr.org/2020/819

Thanks Eysa Lee for the artwork!