Non-Interactive Secure Computation (NISC)

Goal: receiver gets $f(x, y)$ for a public function $f$. 

Non-Interactive Secure Computation (NISC)

Goal: receiver gets $f(x, y)$ for a public function $f$. 
Non-Interactive Secure Computation (NISC)

E.g. FHE => Semi-honest NISC

Goal: receiver gets $f(x, y)$ for a public function $f$. 
Non-Interactive Secure Computation (NISC)

\[ f(x, y) \]

**Goal:** receiver gets \( f(x, y) \) for a public function \( f \).
Non-Interactive Secure Computation (NISC)

E.g. FHE $\iff$ Semi-honest NISC

Goal: receiver gets $f(x, y)$ for a public function $f$. 

$S$  \hspace{1cm} $R$

$y$  \hspace{2cm} $x$

$f(x, y)$
Non-Interactive Secure Computation (NISC)

E.g. FHE $\implies$ Semi-honest NISC

Goal: receiver gets $f(x, y)$ for a public function $f$. 
Non-Interactive Secure Computation (NISC)

E.g. FHE $\rightarrow$ Semi-honest NISC

Goal: receiver gets $f(x, y)$ for a public function $f$. 

\[ \text{Enc}(x), \text{Enc}(f(x, y)) \]

\[
\begin{array}{c}
S \\
y
\end{array} \quad \leftrightarrow \quad \begin{array}{c}
\text{Enc}(x) \\
\text{Enc}(f(x, y))
\end{array} \quad \begin{array}{c}
R \\
x
\end{array}
\]

f(x, y)
Garbled Circuit + OT $\implies$ Semi-honest NISC [Kilian’88]
Garbled Circuit + OT $\Rightarrow$ Semi-honest NISC [Kilian’88]

$\tilde{C}$ and tags

<table>
<thead>
<tr>
<th>$w_{1,0}$</th>
<th>$w_{1,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_{2,0}$</td>
<td>$w_{2,1}$</td>
</tr>
<tr>
<td>$w_{3,0}$</td>
<td>$w_{3,1}$</td>
</tr>
<tr>
<td>$w_{4,0}$</td>
<td>$w_{4,1}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$w_{n,0}$</td>
<td>$w_{n,1}$</td>
</tr>
</tbody>
</table>
Garbled Circuit + OT $\Rightarrow$ Semi-honest NISC [Kilian’88]

$S$ and $R$

$\tilde{C}$ and tags

$$
\begin{array}{cc}
  w_{1,0} & w_{1,1} \\
  w_{2,0} & w_{2,1} \\
  w_{3,0} & w_{3,1} \\
  w_{4,0} & w_{4,1} \\
  \vdots \\
  w_{n,0} & w_{n,1}
\end{array}
$$

$x = \begin{bmatrix}
  1 \\
  0 \\
  0 \\
  1 \\
  \vdots \\
  1
\end{bmatrix}$
Garbled Circuit + OT $\implies$ Semi-honest NISC [Kilian’88]

$$S \quad \begin{array}{c} y \\ \end{array} \quad R \quad \begin{array}{c} x \\ \end{array}$$

\[ x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \]

\[ \tilde{C} \text{ and tags} \]

<table>
<thead>
<tr>
<th></th>
<th>( w_{1,0} )</th>
<th>( w_{1,1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( w_{2,0} )</td>
<td>( w_{2,1} )</td>
</tr>
<tr>
<td></td>
<td>( w_{3,0} )</td>
<td>( w_{3,1} )</td>
</tr>
<tr>
<td></td>
<td>( w_{4,0} )</td>
<td>( w_{4,1} )</td>
</tr>
<tr>
<td></td>
<td>\vdots</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( w_{n,0} )</td>
<td>( w_{n,1} )</td>
</tr>
</tbody>
</table>
Garbled Circuit $+$ OT $\implies$ Semi-honest NISC [Kilian’88]

$\tilde{C}$ and tags

$\tilde{C}$ and $(w_i,x_i)_{i=1}^n$ reveals $f(x,y)$ and nothing else computationally.

$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$
Garbled Circuit + OT $\implies$ Semi-honest NISC [Kilian'88]

$S \xrightarrow{y} \tilde{C} \xrightarrow{x} R$

$\tilde{C}$ and tags $w_i, 0, w_i, 1$

$\tilde{C}$ and $(w_i, x_i)_{i=1}^n$ reveals $f(x, y)$ and nothing else computationally.
Garbled Circuit + OT ⇒ Semi-honest NISC [Kilian’88]

\[ \tilde{C} \quad \text{AND tags} \]

\[ w_{i,0}, w_{i,1} \quad \text{OT} \quad x_i \quad w_{i,x_i} \quad x \]

\[ \tilde{C} \text{ and } (w_{i,x_i})_{i=1}^n \text{ reveals } f(x, y) \text{ and nothing else computationally.} \]
NISC in OT-hybrid model

Advantages
- OT realization from various models/assumptions
- Efficiency
  - Malicious Security [Ishai-Kushilevitz-Ostrovsky-Prabhakaran-Sahai’88]
    - Information-theoretical NISC for NC$^0$ in OT-hybrid.
    - NISC in OT-hybrid using black-box PRG.

Disadvantages
- NOT reusable secure.
NISC in OT-hybrid model

Advantages

▶ OT realization from various models/assumptions
▶ Efficiency
▶ Malicious Security [Ishai-Kushilevitz-Ostrovsky-Prabhakaran-Sahai’88]
  ▶ Information-theoretical NISC for $\textbf{NC}^0$ in OT-hybrid.
  ▶ NISC in OT-hybrid using black-box PRG.

Disadvantages

▶ NOT reusable secure.
NISC in OT-hybrid model

Advantages
- OT realization from various models/assumptions
- Efficiency
- Malicious Security [Ishai-Kushilevitz-Ostrovsky-Prabhakaran-Sahai’88]
  - Information-theoretical NISC for $\text{NC}^0$ in OT-hybrid.
  - NISC in OT-hybrid using black-box PRG.

Disadvantages
- NOT reusable secure.
Reusable NISC

$S \ y$

$R \ x$
Reusable NISC

“encryption” of my data $C_x$
Reusable NISC

“encryption” of my data $C_x$
Reusable NISC

“encryption” of my data $C_x$

$S \xrightarrow{y} \text{msg} \xrightarrow{x} R$

$f(x, y)$
Reusable NISC

“encryption” of my data $C_x$

$S$  $\xrightarrow{\text{msg}}$  $R$

$S$  $\xrightarrow{msg}$  $R$

$f(x, y)$

Reusability: Safe for receiver to reuse first msg and randomness
Reusable NISC

S

y

S'

y', y''

msg

R

x

"encryption" of my data $C_x$

**Reusability**: Safe for receiver to reuse first msg and randomness

$f(x, y)$
Reusable NISC

“encryption” of my data $C_x$

Reusability: Safe for receiver to reuse first msg and randomness
Reusable NISC

"encryption" of my data $C_x$

Reusability: Safe for receiver to reuse first msg and randomness
Reusable NISC

“encryption” of my data $C_x$

Reusability: Safe for receiver to reuse first msg and randomness
NISC in OT-hybrid model

\[ S \xrightarrow{y} \tilde{C} \xrightarrow{\text{reusable OT}} R \]

\[ w_{i,0}, w_{i,1} \]

\[ \tilde{C} \text{ and tags} \]

\[
\begin{array}{cc}
  w_{1,0} & w_{1,1} \\
  w_{2,0} & w_{2,1} \\
  w_{3,0} & w_{3,1} \\
  w_{4,0} & w_{4,1} \\
  \vdots \\
  w_{n,0} & w_{n,1} \\
\end{array}
\]

\[ x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \]
NISC in OT-hybrid model

\[ S \xrightarrow{y} \tilde{C} \xrightarrow{w_i,0, w_i,1} \text{reusable OT} \xrightarrow{x_i} \tilde{C} \xrightarrow{w_i, x_i} R \]

\[ \begin{array}{c|c}
    w_{1,0} & \text{mess} \\
    w_{2,0} & w_{2,1} \\
    w_{3,0} & w_{3,1} \\
    w_{4,0} & w_{4,1} \\
    \vdots & \vdots \\
    w_{n,0} & w_{n,1} \\
\end{array} \]

\[ x = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \]
NISC in OT-hybrid model

\[ w_{i,0}, w_{i,1} \]

\[ \tilde{\mathcal{C}} \]

\[ x_i \]

\[ w_{i,x_i} \]

\[ x \]

\[ x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \]

\[ w_{1,0} \text{ mess} \]

\[ w_{2,0} \quad w_{2,1} \]

\[ w_{3,0} \quad w_{3,1} \]

\[ w_{4,0} \quad w_{4,1} \]

\[ \vdots \]

\[ w_{n,0} \quad w_{n,1} \]

Replacing \( w_{1,1} \) changes \( \tilde{\mathcal{C}} \)'s behaviour \[ \implies \]

\( x[1] = 1 \)

thus **NO security** against malicious sender.
NISC in OT-hybrid model

\[ y \xrightarrow{w_{i,0}, w_{i,1}} \tilde{C} \xrightarrow{x_i} x \]

\[ \tilde{C} \text{ and tags} \]

<table>
<thead>
<tr>
<th>( w_{i,0} )</th>
<th>( w_{i,1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>mess</td>
<td></td>
</tr>
<tr>
<td>( w_{2,0} )</td>
<td>( w_{2,1} )</td>
</tr>
<tr>
<td>( w_{3,0} )</td>
<td>( w_{3,1} )</td>
</tr>
<tr>
<td>( w_{4,0} )</td>
<td>( w_{4,1} )</td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
</tr>
<tr>
<td>( w_{n,0} )</td>
<td>( w_{n,1} )</td>
</tr>
</tbody>
</table>

\[ x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \]
NISC in OT-hybrid model + one-shot UC-security [IKOPS’11]

\[ \tilde{C} \xleftarrow{\text{encoding } \tilde{x}} \tilde{C} \xrightarrow{\text{OT input be encoding } \tilde{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix}} \]
NISC in OT-hybrid model + one-shot UC-security [IKOPS’11]

\[ y \xrightarrow{w_{i,0}, w_{i,1}} \tilde{C} \xrightarrow{\tilde{x}_i, w_{i,0}} x \]

\[ \tilde{C} \text{ and tags} \]

| \( w_{1,0} \) | \text{mess} |
| \( w_{2,0} \) | \( w_{2,1} \) |
| \( w_{3,0} \) | \( w_{3,1} \) |
| \( w_{4,0} \) | \( w_{4,1} \) |
| \vdots |
| \( w_{n,0} \) | \( w_{n,1} \) |

let OT input be encoding \( \tilde{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \)

A few bits of \( \tilde{x} \) leaks no information about \( x \).
NISC in OT-hybrid model + one-shot UC-security [IKOPS’11]

\[ \tilde{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \]

Repeat the attack to learn the whole encoding $\tilde{x}$ thus NO reusable security against malicious sender.
Our Results

Impossible to patch the protocol against malicious adversaries in reusable settings, as we show...

Theorem 1

There is no information-theoretic reusable NISC in rOT-hybrid model.
Our Results

Impossible to patch the protocol against malicious adversaries in reusable settings, as we show...

Theorem 1

There is no information-theoretic reusable NISC in rOT-hybrid model.

There is no reusable NISC for certain functionalities in rOT-hybrid model with black-box simulation, assuming OWF.
Our Results

Impossible to patch the protocol against malicious adversaries in reusable settings, as we show...

Theorem 1

There is no information-theoretic reusable NISC in rOT-hybrid model.

There is no reusable NISC for certain functionalities in rOT-hybrid model with black-box simulation, assuming OWF.

Expansive alternative:
Semi-honest NISC + reusable NIZK $\implies$ reusable NISC.
Our Results (continue)

NEW primitive: Oblivious linear function evaluation (OLE)

\[ S \xrightarrow{\text{OLE}} R \]

\[ a, b \in \mathbb{F} \quad x \in \mathbb{F} \]

Theorem 2
An information-theoretical UC-secure reusable NISC protocol in \textit{rOLE}-hybrid model.

Theorem 3
An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.

Security loss \( \approx \frac{1}{|\mathbb{F}|} \)
NEW primitive: Oblivious linear function evaluation (OLE)

\[ S, \quad a, b \in \mathbb{F} \quad \text{get} \quad ax + b \in \mathbb{F} \]

Degenerate into OT if \(|\mathbb{F}| = 2\).

Theorem 2
An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.

Theorem 3
An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.
Our Results (continue)

NEW primitive: Oblivious linear function evaluation (OLE)

Theorem 2
An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.
Our Results (continue)

NEW primitive: Oblivious linear function evaluation (OLE)

\[ S \xrightarrow{a, b \in \mathbb{F}} R \xleftarrow{x \in \mathbb{F}} ax + b \in \mathbb{F} \]

Degenerate into OT if \(|\mathbb{F}| = 2\).

**Theorem 2**
An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.

**Theorem 3**
An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.
NEW primitive: Oblivious linear function evaluation (OLE)

An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.

Theorem 2

Theorem 3

An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.
NEW primitive: Oblivious linear function evaluation (OLE)

\[ S \xrightarrow{a, b \in \mathbb{F}} R \xrightarrow{x \in \mathbb{F}} \text{get } ax + b \in \mathbb{F} \]

Degenerate into OT if \(|\mathbb{F}| = 2\).

**Theorem 2**
An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.

Security loss \( \approx \frac{1}{|\mathbb{F}|} \)

**Theorem 3**
An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.
How to Lift One-shot Security to Reusability

\[
S \xrightarrow{a_i, b_i} \text{rOLE} \xrightarrow{\tilde{x}_i} R \quad a_i\tilde{x}_i + b_i \\
\]

UC-security: \(\exists\) an efficient simulator \(S(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \)

No Abort (optional): When abnormal behavior was detected, output \(f(x, 0)\)

Difficulty: distribution \(y^* = \Rightarrow f(x, y^*)\) has entropy in ideal world \(\Rightarrow\) leak information of receiver's randomness in real world

"Strong" UC-security = \(\Rightarrow\) Reusability

The simulator is deterministic
How to Lift One-shot Security to Reusability

UC-security: \( \exists \) an efficient simulator \( S(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \)

- No Abort (optional): When abnormal behavior was detected, output \( f(x, 0) \)
- Difficulty: distribution \( y^* = f(x, y^*) \) has entropy in ideal world \( \Rightarrow \) leak information of receiver's randomness in real world


"Strong" UC-security \( \Rightarrow \) Reusability

The simulator is deterministic
How to Lift One-shot Security to Reusability

UC-security: $\exists$ an efficient simulator $S(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^*$

No Abort (optional): When abnormal behavior was detected, output $f(x, 0)$

Difficulty: distribution $y^* = f(x, y^*)$ has entropy in ideal world $\Rightarrow$ leak information of receiver's randomness in real world

"Strong" UC-security $\Rightarrow$ Reusability

The simulator is deterministic
How to Lift One-shot Security to Reusability

UC-security: \( \exists \) an efficient simulator \( \mathcal{S} \)
\( \mathcal{S}(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \)
How to Lift One-shot Security to Reusability

- **UC-security**: ∃ an efficient simulator $S$
  $S(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^*$

- **No Abort** (optional): When abnormal behavior was detected, output $f(x, 0)$
How to Lift One-shot Security to Reusability

- **UC-security**: \( \exists \) an efficient simulator \( \mathcal{S} \)
  
  \[ \mathcal{S}(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \]

- **No Abort** (optional): When abnormal behavior was detected, output \( f(x, 0) \)

- **Difficulty**: distribution \( y^* \) \( \Rightarrow \) \( f(x, y^*) \) has entropy in ideal world
  
  \( \Rightarrow \) leak information of receiver’s randomness in real world

\[ f(x, y^*) \]
How to Lift One-shot Security to Reusability

UC-security: \( \exists \) an efficient simulator \( \mathcal{I} \)
\( \mathcal{I}(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \)

No Abort (optional): When abnormal behavior was detected, output \( f(x, 0) \)

Difficulty: distribution \( y^* \) \( \Rightarrow \) \( f(x, y^*) \) has entropy in ideal world
\( \Rightarrow \) leak information of receiver’s randomness in real world
How to Lift One-shot Security to Reusability

UC-security: \( \exists \) an efficient simulator \( \mathcal{S} \)
\( \mathcal{S}(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \)

No Abort (optional): When abnormal behavior was detected, output \( f(x, 0) \)

Difficulty: distribution \( y^* \) \( \rightarrow \) \( f(x, y^*) \) has entropy in ideal world
\( \rightarrow \) leak information of receiver’s randomness in real world

The simulator is deterministic
How to Lift One-shot Security to Reusability

\[ S \xrightarrow{a_i, b_i} \text{rOLE} \xleftarrow{\tilde{x}_i} a_i \tilde{x}_i + b_i \rightarrow R \]

- **UC-security**: \( \exists \) an efficient simulator \( \mathcal{I} \)
  \( \mathcal{I}(a_1, b_1, a_2, b_2, \ldots) \rightarrow y^* \)

- **No Abort** (optional): When abnormal behavior was detected, output \( f(x, 0) \)

- **Difficulty**: distribution \( y^* \) \( \Rightarrow \) \( f(x, y^*) \) has entropy in ideal world
  \( \Rightarrow \) leak information of receiver’s randomness in real world

- **“Strong” UC-security** \( \Rightarrow \) Reusability
  The simulator is deterministic
Overview: rNISC in rOLE-hybrid model

Assume $f$ is an arithmetic $\text{NC}^1$ circuit or an arithmetic branching program over $\mathbb{F}$

[$\text{IK'}02, \text{AIK'}14$] encode $y \mapsto (A, b)$ s.t. $Ax + b$ reveals $f(x, y)$ and nothing else

Against malicious sender: detect if $(A, b)$ is honestly generated, i.e. satisfies some simple arithmetic constraints

Certified rOLE $\rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases}$
Overview: rNISC in rOLE-hybrid model

\[
S \quad y \in \mathbb{F}^n
\]

\[
R \quad x \in \mathbb{F}^n
\]

- Assume \( f \) is an arithmetic \( \mathbf{NC}^1 \) circuit or an arithmetic branching program over \( \mathbb{F} \)

- [IK’02,AIK’14] encode \( y \mapsto (A, b) \)
  such that \( Ax + b \) reveals \( f(x, y) \) and nothing else

- Against malicious sender: detect if \( (A, b) \) is honestly generated, i.e. satisfies some simple arithmetic constraints

Certified rOLE \( \rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases} \)
Overview: rNISC in rOLE-hybrid model

Assume \( f \) is an arithmetic \( \text{NC}^1 \) circuit or an arithmetic branching program over \( \mathbb{F} \)

[IK'02, AIK'14] encode \( y \mapsto (A, b) \)
s.t. \( Ax + b \) reveals \( f(x, y) \) and nothing else

Against malicious sender: detect if \((A, b)\) is honestly generated, i.e. satisfies some simple arithmetic constraints

Certified rOLE \( \rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases} \)
Overview: rNISC in rOLE-hybrid model

Assume $f$ is an arithmetic $\text{NC}^1$ circuit or an arithmetic branching program over $\mathbb{F}$

[IK’02, AIK’14] encode $y \mapsto (A, b)$ s.t. $Ax + b$ reveals $f(x, y)$ and nothing else

Against malicious sender: detect if $(A, b)$ is honestly generated, i.e. satisfies some simple arithmetic constraints

Certified rOLE $\rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases}$
Overview: rNISC in rOLE-hybrid model

- Assume $f$ is an arithmetic $\textbf{NC}^1$ circuit or an arithmetic branching program over $\mathbb{F}$
- [IK’02, AIK’14] encode $y \mapsto (A, b)$ s.t. $Ax + b$ reveals $f(x, y)$ and nothing else
- **Against malicious sender:** detect if $(A, b)$ is honestly generated, i.e. satisfies some simple arithmetic constraints

Certified rOLE $\rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases}$
Overview: rNISC in rOLE-hybrid model

Assume $f$ is an arithmetic $\textbf{NC}^1$ circuit or an arithmetic branching program over $\mathbb{F}$

[IK’02, AIK’14] encode $y \mapsto (A, b)$ s.t. $Ax + b$ reveals $f(x, y)$ and nothing else

Against malicious sender: detect if $(A, b)$ is honestly generated, i.e. satisfies some simple arithmetic constraints

$\text{Certified rOLE} \rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases}$
Overview: rNISC in rOLE-hybrid model

Assume $f$ is an arithmetic $\textbf{NC}^1$ circuit or an arithmetic branching program over $\mathbb{F}$

[IK’02, AIK’14] encode $y \mapsto (A, b)$ s.t. $Ax + b$ reveals $f(x, y)$ and nothing else

Against malicious sender: detect if $(A, b)$ is honestly generated, i.e. satisfies some simple arithmetic constraints

Certified rOLE $\rightarrow \begin{cases} Ax + b, & \text{if } (A, b) \text{ satisfies constraints} \\ \bot, & \text{otherwise} \end{cases}$
Certified rOLE

$S \quad R$

Sender can prove $(a_1, b_1, a_2, b_2, \ldots)$ satisfies arithmetic constraints

Side product: reusable DV-NIZK in rOLE-hybrid model.
Certified rOLE

Sender can prove \((a_1, b_1, a_2, b_2, \ldots)\) satisfies arithmetic constraints

Side product: reusable DV-NIZK in rOLE-hybrid model.
Certified rOLE

Sender can prove \((a_1, b_1, a_2, b_2, \ldots)\) satisfies arithmetic constraints

Side product: reusable DV-NIZK in rOLE-hybrid model.
Sender can prove \((a_1, b_1, a_2, b_2, \ldots)\) satisfies arithmetic constraints.
Certified rOLE

Sender can prove \((a_1, b_1, a_2, b_2, \ldots)\) satisfies arithmetic constraints

Side product: reusable DV-NIZK in rOLE-hybrid model.
Certified rOLE

Sender can prove \((a_1, b_1, a_2, b_2, \ldots)\) satisfies arithmetic constraints.

Side product: reusable DV-NIZK in rOLE-hybrid model.
Certified rOLE

Sender can prove \((a_1, b_1, a_2, b_2, \ldots)\) satisfies arithmetic constraints \(a_i = a_j\) for some \((i, j)\)

Side product: reusable DV-NIZK in rOLE-hybrid model.
Certified rOLE

\[ S \]

\[ \text{Certified rOLE} \]

\[ \vdots \]

\[ \text{Certified rOLE} \]

\[ R \]
Certified rOLE

\[ w \leftarrow F \]
Certified rOLE

\[ S \xrightarrow{r \leftarrow F} a, r \xrightarrow{\text{rOLE}} w \xrightarrow{aw + r} R \xrightarrow{w \leftarrow F} \]
Certified rOLE

\[ r \leftarrow F \]

\[ w \leftarrow F \]

\[ aw + r \]

\[ \text{Commitment}(a) \]

\[ a, r \rightarrow \text{rOLE} \]

\[ w \rightarrow \text{Certified rOLE} \]
Certified rOLE

\[ r \leftarrow \mathbb{F} \]

\[ a, r \rightarrow \text{rOLE} \]

\[ w \rightarrow \text{rOLE} \]

\[ aw + r \rightarrow w \leftarrow \mathbb{F} \]

\[ ax_i + b \]

\[ \text{Target} \]

\[ aw + r \]

\[ \text{Commitment}(a) \]
Certified rOLE

\[ r \leftarrow F \]

\[ a, r \rightarrow \text{rOLE} \]

\[ w \rightarrow \text{rOLE} \]

\[ aw + r \rightarrow \text{rOLE} \]

\[ \hat{x}_i \rightarrow \text{rOLE} \]

\[ \hat{x}_i = x_i - w \hat{x}_i \]

\[ ax_i + b \rightarrow \text{Target} \]

\[ aw + r \rightarrow \text{Commitment}(a) \]
Certified rOLE

\[ r \leftarrow F \]
\[ r' \leftarrow F \]

\[ r, r' \]
\[ r\hat{x} + r' \]
\[ a, b + r' \]

\[ \hat{x}_i = x_i - w\hat{x}_i \]
\[ a\hat{x}_i + b + r' \]

\[ ax_i + b \]
\[ aw + r \]

Target

Commitment(a)
Certified rOLE

\[ r \leftarrow F \]
\[ r' \leftarrow F \]

\[ a, r \rightarrow w \rightarrow \text{rOLE} \rightarrow aw + r \rightarrow R \]
\[ a, b + r' \rightarrow \text{rOLE} \rightarrow \hat{x}_i \rightarrow r\hat{x}_i + r' \rightarrow w \leftarrow F \]
\[ \hat{x}_i \leftarrow F \]

\[ \hat{x}_i = x_i - w\hat{x}_i \]
\[ a\hat{x}_i + b + r' \rightarrow \vdots \]

Target
Commitment\((a)\)
rOLE outputs
Certified rOLE

\[ ax_i + b = (aw + r) \cdot \hat{x}_i - (r\hat{x}_i + r') + (a\hat{x}_i + b + r') \]

Target

Commitment(a)

rOLE outputs
Certified rOLE

\[ a, r \rightarrow \text{rOLE} \]
\[ w \rightarrow \text{rOLE} \]
\[ r, r' \rightarrow \text{rOLE} \]
\[ \hat{x}_i \rightarrow w, \hat{x}_i \leftarrow \mathbb{F} \]

\[ a, b + r' \rightarrow \text{rOLE} \]
\[ \hat{x}_i = x_i - w\hat{x}_i \rightarrow \]

\[ ax_i + b = (aw + r)\cdot\hat{x}_i - (r\hat{x}_i + r') + e(a\hat{x}_i + b + r') \]

Target
Commitment\((a)\)
rOLE outputs

Target Commitment
Correctness: Above equation.
UC-secure against Receiver:
\[ x_i := w\hat{x}_i + \hat{x}_i \]
"Strong" UC-secure against Sender:
\[ \text{Deviate} \Rightarrow \text{random output} \]
Certified rOLE

\[ \hat{x}_i = x_i - w \hat{x}_i \]

\[ ax_i + b = (aw + r) \cdot \hat{x}_i - (r \hat{x}_i + r') + e(a \hat{x}_i + b + r') \]

Certified rOLE outputs

Correctness: Above equation.
Certified rOLE

S → rOLE → R

\[ r, r' \leftarrow \mathbb{F} \]

\[ a, r \]

\[ r, r' \]

\[ a, b + r' \]

\[ x_i = w\hat{x}_i + \hat{x}_i \]

\[ ax_i + b = (aw + r) \cdot \hat{x}_i - (r\hat{x}_i + r') + e(a\hat{x}_i + b + r') \]

Target

Commitment(a)

rOLE outputs

- Correctness: Above equation.
- UC-secure against Receiver: \( x_i := w\hat{x}_i + \hat{x}_i \).
Certified rOLE

\[ S \xrightarrow{r, r'} F \]

\[ \hat{x}_i \leftarrow F \]

\[ a, b + r' \rightarrow \text{rOLE} \]

\[ \hat{x}_i = x_i - w\hat{x}_i \]

\[ ax_i + b = (aw + r) \cdot \hat{x}_i - (r\hat{x}_i + r') + e(a\hat{x}_i + b + r') \]

Target  Commitment(a)  rOLE outputs

- Correctness: Above equation.
- UC-secure against Receiver: \( x_i := w\hat{x}_i + \hat{x}_i \).
- "Strong" UC-secure against Sender:
Certified rOLE

\[
\begin{align*}
r, r' &\leftarrow \mathbb{F} \\
a, b + r' &\rightarrow \text{rOLE} \\
\hat{x}_i &\leftarrow \text{rOLE} \\
\hat{x}_i = x_i - w\hat{x}_i &\rightarrow \\
a x_i + b &= (aw + r) \cdot \hat{x}_i - (r\hat{x}_i + r') + e(a\hat{x}_i + b + r') \\
\end{align*}
\]

- **Target**: \(a x_i + b\)
- **Commitment** \((a)\)**: \(\hat{x}_i = x_i - w\hat{x}_i\)
- **rOLE outputs**: \(w, \hat{x}_i \leftarrow \mathbb{F}\)

- **Correctness**: Above equation.
- **UC-secure against Receiver**: \(x_i := w\hat{x}_i + \hat{x}_i\).
- **“Strong” UC-secure against Sender**: 
Certified rOLE

\[ r, r' \leftarrow \mathbb{F} \]

\[ a, b + r' \rightarrow \]

\[ a, \hat{x}_i \rightarrow \]

\[ \hat{x}_i = x_i - w\hat{x}_i \]

\[ aw + r \cdot \hat{x}_i - (r\hat{x}_i + r') + e(a\hat{x}_i + b + r') \]

Target

Commitment(a)

rOLE outputs

- Correctness: Above equation.
- UC-secure against Receiver: \( x_i := w\hat{x}_i + \hat{x}_i \).
- “Strong” UC-secure against Sender: Deviate \( \implies \) random output.
Certified rOLE

\[ r, r' \leftarrow \mathbb{F} \]

\[ a, b + r' \leftarrow \mathbb{F} \]

\[ a, r \]

\[ r, r' \]

\[ w, \hat{x}_i \leftarrow \mathbb{F} \]

\[ \hat{x}_i = x_i - w\hat{x}_i \]

\[ ax_i + b = (aw + r) \cdot \hat{x}_i - (r\hat{x}_i + r') + e(a\hat{x}_i + b + r') \]

- **Correctness**: Above equation.
- **UC-secure against Receiver**: \( x_i := w\hat{x}_i + \hat{x}_i \).
- **“Strong” UC-secure against Sender**: Deviate \( \Rightarrow \) random output not yet
Our Results

NEW primitive: Oblivious linear function evaluation (OLE)

Theorem 2
An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.

Theorem 3
An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.
Our Results

NEW primitive: Oblivious linear function evaluation (OLE)

\[ S \xrightarrow{\text{get}} ax + b \in \mathbb{F} \]

**Theorem 2**
An information-theoretical UC-secure reusable NISC protocol in rOLE-hybrid model.

**Theorem 3**
An UC-secure 2-msg reusable OLE protocol in the CRS setting, under Paillier assumption.
rOLE from Paillier

**Dual-mode** (similar to OT from [PVW’08])

**Mode I**

\[ S \leftarrow a, b \]
\[ \text{crs} \leftarrow D_1 \]
\[ x \]

**Mode II**

\[ S \leftarrow a, b \]
\[ \text{crs} \leftarrow D_2 \]
\[ x \]
rOLE from Paillier

Dual-mode (similar to OT from [PVW’08])

Mode I

\[ S \leftarrow a, b \]
\[ \text{crs} \leftarrow D_1 \]
\[ x \leftarrow \text{Enc}(x) \]

Mode II

\[ S \leftarrow a, b \]
\[ \text{crs} \leftarrow D_2 \]

Efficient simulator against unbounded malicious receiver

Efficient simulator against unbounded malicious sender
rOLE from Paillier

Dual-mode (similar to OT from [PVW’08])

Mode I

\( S \)

\( a, b \)

\( x \)

\( \text{crs} \leftarrow \mathcal{D}_1 \)

\( \text{Enc}(x) \)

\( \text{Enc}(a - r) \)

\( \text{Enc}(b + rx) \)

Mode II

\( S \)

\( a, b \)

\( x \)

\( \text{crs} \leftarrow \mathcal{D}_2 \)

Efficient simulator against unbounded malicious receiver

Efficient simulator against unbounded malicious sender
rOLE from Paillier

**Dual-mode** (similar to OT from [PVW’08])

**Mode I**

- $S \leftarrow a, b$
- $\text{crs} \leftarrow \mathcal{D}_1$
- $R \leftarrow x$
- $\text{Enc}(x)$
- $\text{Enc}(a - r)$
- $\text{Enc}(b + rx)$

Efficient simulator against unbounded malicious receiver

**Mode II**

- $S \leftarrow a, b$
- $\text{crs} \leftarrow \mathcal{D}_2$
- $R \leftarrow x$
- $\text{Enc}(x)$
- $\text{Enc}(a - r)$
- $\text{Enc}(b + rx)$

Efficient simulator against unbounded malicious sender
rOLE from Paillier

**Dual-mode** (similar to OT from [PVW’08])

**Mode I**
- S sends \( a, b \)
- CRS \( \leftarrow D_1 \)
- \( x \) is indistinguishable from \( D_2 \)
- \( x \) is indistinguishable from \( D_1 \)
- Efficient simulator against unbounded malicious receiver

**Mode II**
- S sends \( a, b \)
- CRS \( \leftarrow D_2 \)
- Efficient simulator against unbounded malicious sender

\[
\begin{align*}
S & \quad a, b \\
\text{Sends} & \\
\text{CRS} & \leftarrow D_1 \\
R & \quad x \\
S & \quad a, b \\
\text{Sends} & \\
\text{CRS} & \leftarrow D_2 \\
R & \quad x
\end{align*}
\]
rOLE from Paillier

**Dual-mode** (similar to OT from [PVW’08])

**Mode I**

\[
\begin{align*}
S &\leftarrow a, b \\
\text{crs} &\leftarrow D_1 \\
\text{cr} &\leftarrow \mathbb{D}_1 \\
\text{Enc}(x) &\leftarrow \mathbb{D}_1 \\
\text{Enc}(a - r) &\leftarrow \mathbb{D}_1 \\
\text{Enc}(b + rx) &\leftarrow \mathbb{D}_1 \\
\end{align*}
\]

Efficient simulator against unbounded malicious receiver

**Mode II**

\[
\begin{align*}
S &\leftarrow a, b \\
\text{crs} &\leftarrow D_2 \\
\text{cr} &\leftarrow \mathbb{D}_2 \\
\text{Enc}(0) &\leftarrow \mathbb{D}_2 \\
\text{Enc}(a) &\leftarrow \mathbb{D}_2 \\
\text{Enc}(b) &\leftarrow \mathbb{D}_2 \\
\end{align*}
\]

Efficient simulator against unbounded malicious sender
rOLE from Paillier

Dual-mode (similar to OT from [PVW'08])

Mode I

\[ S \leftarrow a, b \]

\[ \text{crs} \leftarrow \mathcal{D}_1 \]

\[ x \]

\[ \text{Efficient simulator against unbounded malicious receiver} \]

\[ \text{Enc}(x) \]

\[ \text{Enc}(a - r) \]

\[ \text{Enc}(b + rx) \]

Mode II

\[ S \leftarrow a, b \]

\[ \text{crs} \leftarrow \mathcal{D}_2 \]

\[ x \]

\[ \text{Efficient simulator against unbounded malicious sender} \]

\[ \text{Enc}(0) \]

\[ \text{Enc}(a) \]

\[ \text{Enc}(b) \]
rOLE from Paillier

**Dual-mode** (similar to OT from [PVW’08])

\[ \mathcal{D}_1 \text{ is indistinguishable from } \mathcal{D}_2 \]

**Mode I**

- **S**
  - \( a, b \)
- **R**
  - \( x \)
- **crs** \( \leftarrow \mathcal{D}_1 \)
- **Efficient simulator against unbounded malicious receiver**
- \( \text{Enc}(x) \)
- \( \text{Enc}(a - r) \)
- \( \text{Enc}(b + rx) \)

**Mode II**

- **S**
  - \( a, b \)
- **R**
  - \( x \)
- **crs** \( \leftarrow \mathcal{D}_2 \)
- **Efficient simulator against unbounded malicious sender**
- \( \text{Enc}(0) \)
- \( \text{Enc}(a) \)
- \( \text{Enc}(b) \)
Paillier Encryption Scheme

KeyGen $\rightarrow$ public key, trapdoor
Paillier Encryption Scheme

KeyGen → public key, trapdoor

Encrypt randomness \( r \) → Enc\(_r\)(x) → Decrypt trapdoor → x

\[ \text{Encrypt} \ x \text{ randomness} \ r \text{ Decrypt} \ x \]
Paillier Encryption Scheme

KeyGen $\rightarrow$ public key, trapdoor

Encrypt $\rightarrow$ $\text{Enc}_r(x)$

randomness $r$

Decrypt $\rightarrow$ $x$

$\text{Enc}_0(x) \rightarrow$ Decrypt $\rightarrow$ $x$

$\text{Enc}_r(x) \cdot \text{Enc}_s(y) = \text{Enc}_r(x + y)$
Paillier Encryption Scheme

KeyGen $\rightarrow$ public key, trapdoor

Encrypt $\rightarrow$ $\text{Enc}_r(x)$

randomness $r$

Decrypt $\rightarrow$ $x$

\[
\text{Enc}_0(x) \quad \rightarrow \quad \text{Encrypt} \quad \rightarrow \quad \text{Decrypt} \\
\text{Decrypt} \quad \rightarrow \quad x
\]

\[
\text{Enc}_r(x) \cdot \text{Enc}_s(y) = \text{Enc}_{r+s}(x+y)
\]
rOLE from Paillier

$s, a, b$

$x, R$
rOLE from Paillier

CRS (Mode I)

\[ h = \text{Enc}_0(1) \]
\[ w = \text{Enc}_\alpha(0) \]
\[ W_0 = \text{Enc}_\beta(1) \]
rOLE from Paillier

CRS (Mode I)

\[ h = \text{Enc}_0(1) \]
\[ w = \text{Enc}_\alpha(0) \]
\[ W_0 = \text{Enc}_\beta(1) \]

\[ W_1 = w^sk W_0^x = \text{Enc}_{x\beta + \alpha \cdot sk}(x) \]

"Strong" UC-security requires a mechanism to detect malicious sender
rOLE from Paillier

CRS (Mode I)

\[ h = \text{Enc}_0(1) \]
\[ w = \text{Enc}_\alpha(0) \]
\[ W_0 = \text{Enc}_\beta(1) \]

\[ W_1 = w^{sk} W_0^{x} = \text{Enc}_{x\beta + \alpha \cdot sk}(x) \]

\[ \nu = w^{r} = \text{Enc}_{r\alpha}(0) \]

\[ V_0 = h^{a} W_0^{-r} = \text{Enc}_{-r\beta}(a - r) \]

\[ V_1 = h^{b} W_1^{r} = \text{Enc}_{rx\beta + r\alpha \cdot sk}(b + rx) \]

"Strong" UC-security requires a mechanism to detect malicious sender.
rOLE from Paillier

CRS (Mode I)

\[ h = \text{Enc}_0(1) \]
\[ w = \text{Enc}_\alpha(0) \]
\[ W_0 = \text{Enc}_\beta(1) \]

\[ W_1 = w^\text{sk} W_0^x = \text{Enc}_{x\beta + \alpha \cdot \text{sk}}(x) \]

\[ v = w^r = \text{Enc}_{r\alpha}(0) \]

\[ V_0 = h^a W_0^{-r} = \text{Enc}_{-r\beta}(a - r) \]
\[ V_1 = h^b W_1^r = \text{Enc}_{r\beta + r\alpha \cdot \text{sk}}(b + rx) \]

\[ v^{\text{sk}} V_0^x V_1 = \text{Enc}_0(ax + b) \]

"Strong" UC-security requires a mechanism to detect malicious sender.
**rOLE from Paillier**

**RS**

- Sample $a, b$

**S**

- Sample $r$

**CRS (Mode II)**

- $h = \text{Enc}_0(1)$
- $w = \text{Enc}_\alpha(0)$
- $W_0 = \text{Enc}_\beta(0)$

$W_1 = w^sk W_0^x = \text{Enc}_{x\beta+\alpha.sk}(x)$

- $v = w^r = \text{Enc}_{r\alpha}(0)$

- $V_0 = h^a W_0^{-r} = \text{Enc}_{-r\beta}(a - r)$

- $V_1 = h^b W_1^r = \text{Enc}_{rx\beta+r\alpha.sk}(b + rx)$

$v^{sk} V_0^x V_1 = \text{Enc}_0(ax + b)$

"Strong" UC-security requires a mechanism to detect malicious sender.
Strong UC-security requires a mechanism to detect malicious sender.
rOLE from Paillier

CRS (Mode II)

\[ h = \text{Enc}_0(1) \]
\[ w = \text{Enc}_\alpha(0) \]
\[ W_0 = \text{Enc}_\beta(0) \]

\[ W_1 = w^{sk} W_0^x = \text{Enc}_{x\beta + \alpha \cdot sk}(0) \]

\[ v = w^r = \text{Enc}_{r\alpha}(0) \]

\[ V_0 = h^a W_0^{-r} = \text{Enc}_{-r\beta}(a) \]
\[ V_1 = h^b W_1^r = \text{Enc}_{rx\beta + r\alpha \cdot sk}(b) \]

\[ v^{sk} V_0^x V_1 = \text{Enc}_0(ax + b) \]

"Strong" UC-security requires a mechanism to detect malicious sender.
rOLE from Paillier

\begin{align*}
S & \quad \text{sample } a, b \\
R & \quad \text{sample } sk
\end{align*}

\begin{align*}
\text{CRS (Mode II)}
\quad h &= \text{Enc}_0(1) \\
\quad w &= \text{Enc}_\alpha(0) \\
\quad W_0 &= \text{Enc}_\beta(0)
\end{align*}

\begin{align*}
W_1 &= w^{sk} W_0^x = \text{Enc}_{x\beta + \alpha \cdot sk}(0) \\
V_0 &= h^a W_0^{-r} = \text{Enc}_{-r\beta}(a) \\
V_1 &= h^b W_1^r = \text{Enc}_{r\beta + r\alpha \cdot sk}(b)
\end{align*}

\[
v^{sk} V_0^x V_1 = \text{Enc}_0(ax + b)
\]

“Strong” UC-security requires a mechanism to detect malicious sender.
Our Results

- \((\exists \text{ IT rNISC/rOT})\) There is no information-theoretical reusable NISC protocol in rOT-hybrid model.

- \((\text{IT rNISC/rOLE for arithmetic NC}^1)\) Information-theoretical UC-secure reusable NISC protocol for any arithmetic \(\text{NC}^1\) circuit or arithmetic branching program in rOLE-hybrid model.

- \((\text{IT rNIZK/rOLE})\) Information-theoretical UC-secure reusable NIZK protocol in rOLE-hybrid model; \(O(1)\) calls per gate.

- **Previous two + Garbled circuit \(\rightarrow (\text{rNISC/rOLE})\)** UC-secure reusable NISC for general circuits; IT secure against sender; poly\((\lambda)\) calls per gate.

- \((\text{rOLE protocol from Paillier})\) UC-secure reusable 2-message OLE protocol in CRS model; one-side IT secure; c.c. \(O(1)\) group elements per call.
Our Results

- **rNISC** in CRS model assuming the security of Paillier encryption.
- **rNIZK** in CRS model assuming the security of Paillier encryption. c.c. $O(1)$ group elements per gate.
- Statistical designated-verifier NIZK argument for NP in CRS model assuming Paillier.
- Push cryptograph to offline phase.
  - In offline phase: prepare random $((a, b), (x, ax + b))$;
  - In online phase: consume the prepared randomness.
Our Results

▶ **rNISC** in CRS model assuming the security of Paillier encryption.

▶ **rNIZK** in CRS model assuming the security of Paillier encryption. c.c. $O(1)$ group elements per gate.

▶ **Statistical designated-verifier NIZK argument** for NP in CRS model assuming Paillier.

▶ Push cryptograph to offline phase.
   In offline phase: prepare random $((a, b), (x, ax + b))$;
   In online phase: consume the prepared randomness.
Our Results

- **rNISC** in CRS model assuming the security of Paillier encryption.

- **rNIZK** in CRS model assuming the security of Paillier encryption. c.c. $O(1)$ group elements per gate.

- **Statistical designated-verifier NIZK argument** for NP in CRS model assuming Paillier.

- Push cryptograph to offline phase.
  - In offline phase: prepare random $((a, b), (x, ax + b))$;
  - In online phase: consume the prepared randomness.