More Efficient Oblivious Transfer Extensions with Security for Malicious Adversaries

Gilad Asharov
Yehuda Lindell
Thomas Schneider
Michael Zohner

EUROCRYPT 2015

This Talk

This Talk

- Oblivious Transfer Extension

This Talk

- Oblivious Transfer Extension
  - Benny’s talk (Sunday)

This Talk

- Oblivious Transfer Extension
    - Benny’s talk (Sunday)
    - Yehuda’s talk (Monday)

This Talk

- Oblivious Transfer Extension
  - Benny’s talk (Sunday)
  - Yehuda’s talk (Monday)
  - Claudio’s talk (Tuesday)

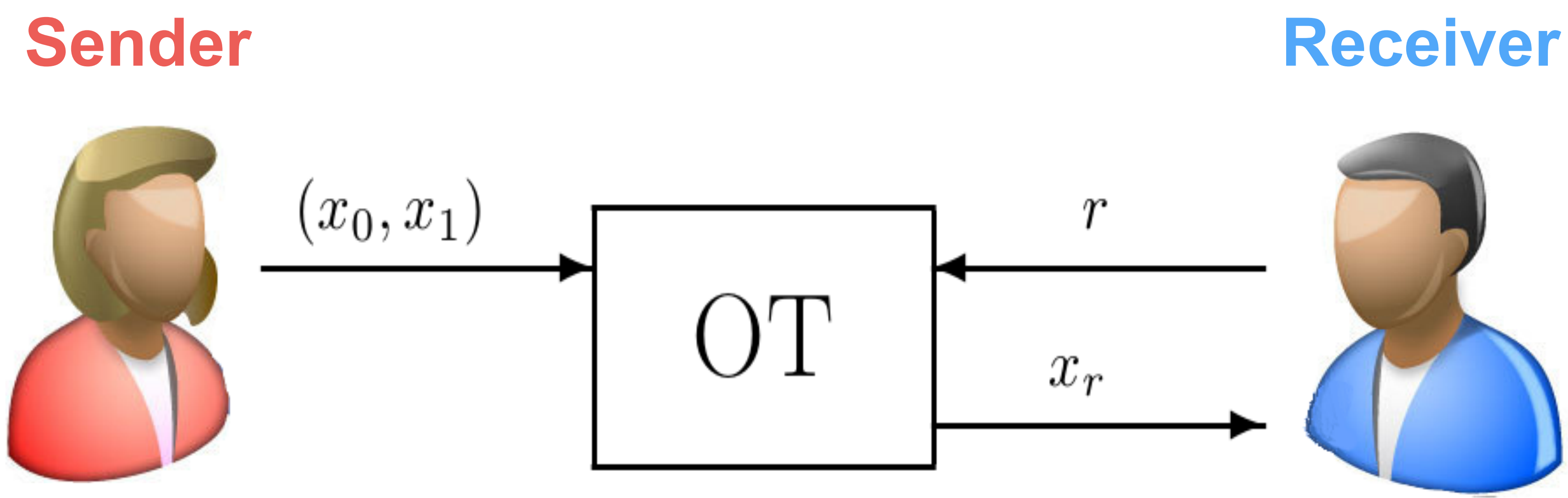
This Talk

- Oblivious Transfer Extension
  - Benny’s talk (Sunday)
  - Yehuda’s talk (Monday)
  - Claudio’s talk (Tuesday)
  - This talk (Thursday)

This Talk

- Oblivious Transfer Extension
  - Benny's talk (Sunday)
  - Yehuda's talk (Monday)
  - Claudio's talk (Tuesday)
  - This talk (Thursday)
- Concrete efficiency in the malicious model
  - Most efficient OT extension protocol, yet
  - Optimized protocol, proofs and implementation

1-out-of-2 Oblivious Transfer



- **INPUT:** **Sender** holds two strings (x_0,x_1), **Receiver** holds r
- **OUTPUT:** **Sender** learns nothing, **Receiver** learns x_r,

Oblivious Transfer and Secure Computation

Oblivious Transfer and Secure Computation

- OT is a basic ingredient in (almost) all protocols for secure computation

Oblivious Transfer and Secure Computation

- OT is a basic ingredient in (almost) all protocols for secure computation

- **Protocols based on Garbled Circuits (Yao):**
  1 OT per ***input***
  [LP07,LPS08,PSSW09,KSS12,FN13,SS13,LR14,HKK+14,FJN14]

Oblivious Transfer and Secure Computation

- OT is a basic ingredient in (almost) all protocols for secure computation
- **Protocols based on Garbled Circuits (Yao):**
  1 OT per ***input***
  [LP07,LPS08,PSSW09,KSS12,FN13,SS13,LR14,HKK+14,FJN14]
- **Protocols based on GMW:**
  1+ OT per ***AND-gate***
  TinyOT [NNOB12,LOS14] MiniMac protocols [DZ13,DLT14]

How Many OT's?

How Many OT's?

- **The AES circuit:** Uses 2^{19} OTs
  (when evaluated with TinyOT)

How Many OT's?

- **The AES circuit:** Uses 2^{19} OTs
  (when evaluated with TinyOT)

- **The PSI circuit:** (for $b=32, n=2^{16}$) Uses 2^{30} OTs
  (when evaluated with TinyOT)

How Many OT's?

- **The AES circuit:** Uses 2^{19} OTs
  (when evaluated with TinyOT)
- **The PSI circuit:** (for $b=32, n=2^{16}$) Uses 2^{30} OTs
  (when evaluated with TinyOT)
- Using [PVW08]: 350 OTs per second

How Many OT's?

- **The AES circuit:** Uses 2^{19} OTs
  (when evaluated with TinyOT)
- **The PSI circuit:** (for $b=32, n=2^{16}$) Uses 2^{30} OTs
  (when evaluated with TinyOT)
- Using [PVW08]: 350 OTs per second
  - 1M (2^{20}) OTs > 45 minutes(!)

How Many OT's?

- **The AES circuit:** Uses 2^{19} OTs
  (when evaluated with TinyOT)
- **The PSI circuit:** (for $b=32, n=2^{16}$) Uses 2^{30} OTs
  (when evaluated with TinyOT)
- Using [PVW08]: 350 OTs per second
  - 1M (2^{20}) OTs > 45 minutes(!)

How Many OT's?

- **The AES circuit:** Uses 2^{19} OTs
  (when evaluated with TinyOT)
- **The PSI circuit:** (for $b=32, n=2^{16}$) Uses 2^{30} OTs
  (when evaluated with TinyOT)
- Using [PVW08]: 350 OTs per second
  - 1M (2^{20}) OTs > 45 minutes(!)
  - 1G (2^{30}) OTs > 45000 minutes > 1 month…

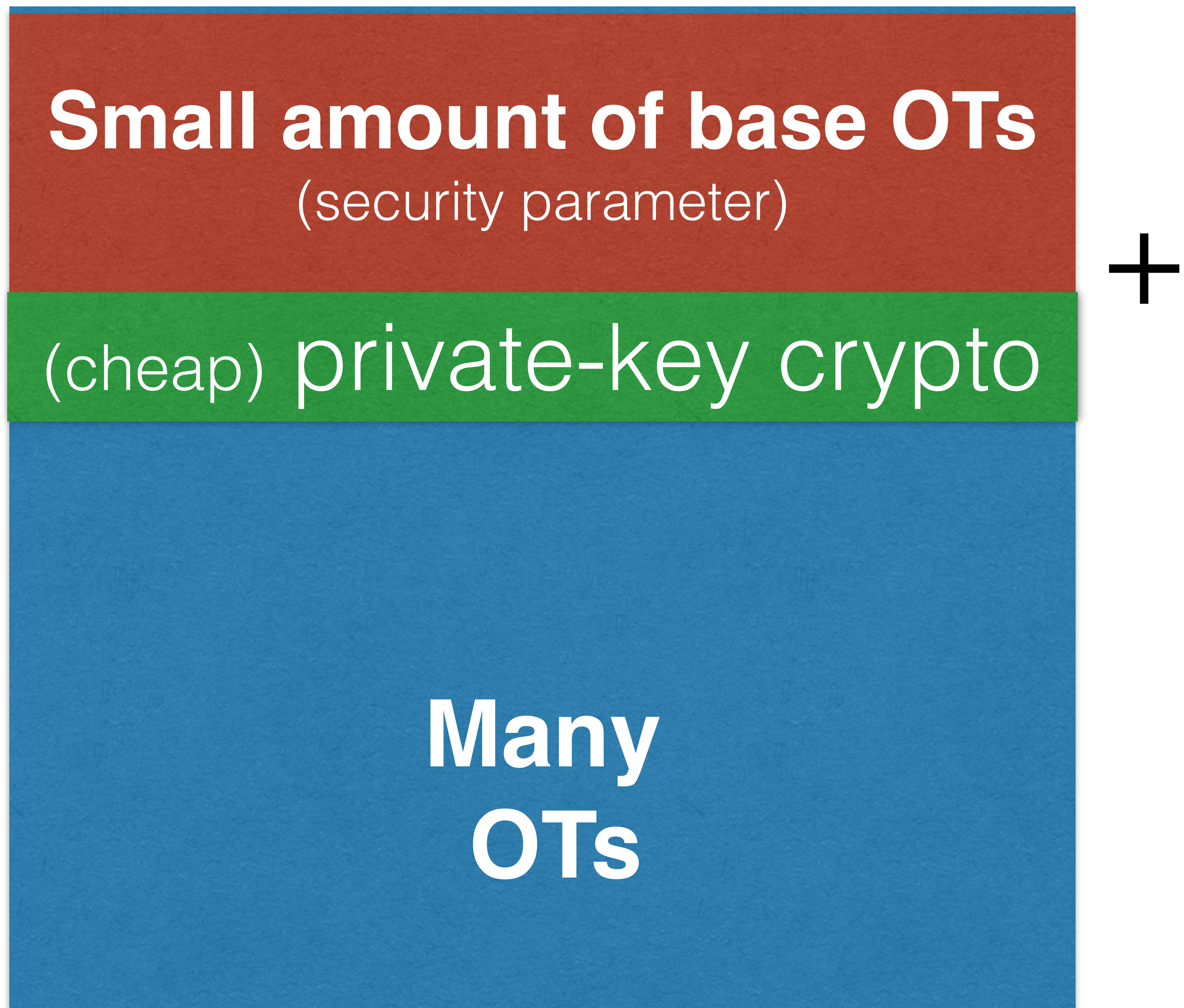
OT Extensions

Small amount of base OTs
(security parameter)

\+

(cheap) private-key crypto

OT Extensions



OT Extension and Related Work

OT Extension and Related Work

- Introduced in [Beaver96]
- Ishai, Kilian, Nissim, Petrank [IKNP03]
  "Extending Oblivious Transfer Efficiently"

OT Extension and Related Work

- Introduced in [Beaver96]
- Ishai, Kilian, Nissim, Petrank [IKNP03]
  "Extending Oblivious Transfer Efficiently"
- Optimizations semi-honest: [KK13, ALSZ13]
- Optimizations malicious:
  [Lar14,NNOB12,HIKN08,Nie07]

Contents

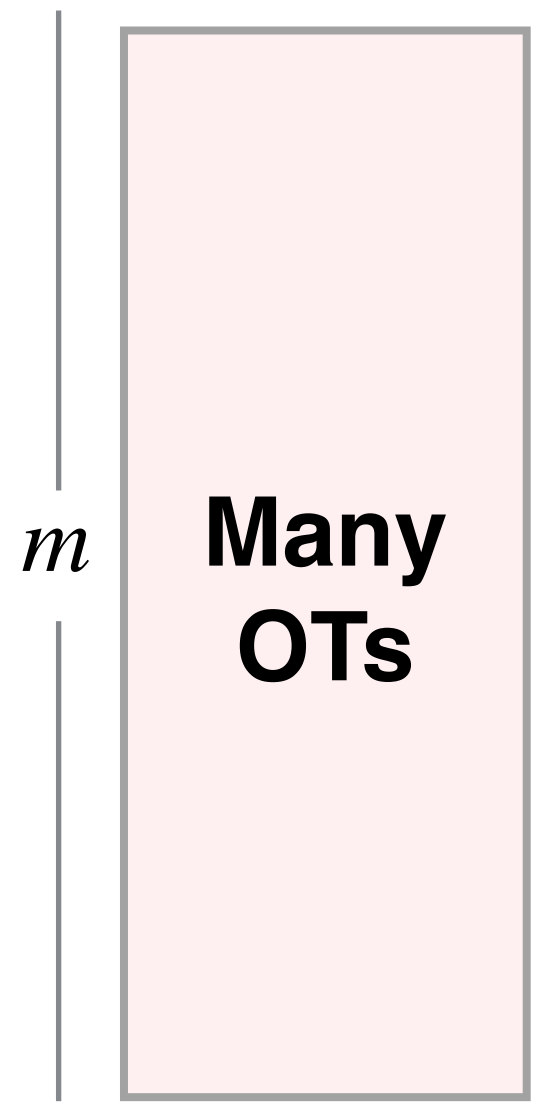
- IKNP protocol
- Our Protocol, Security
- Performance

Extending OT Efficiently[1]

[IKNP03]

[1]Semi-honest

IKNP - Idea



expensive

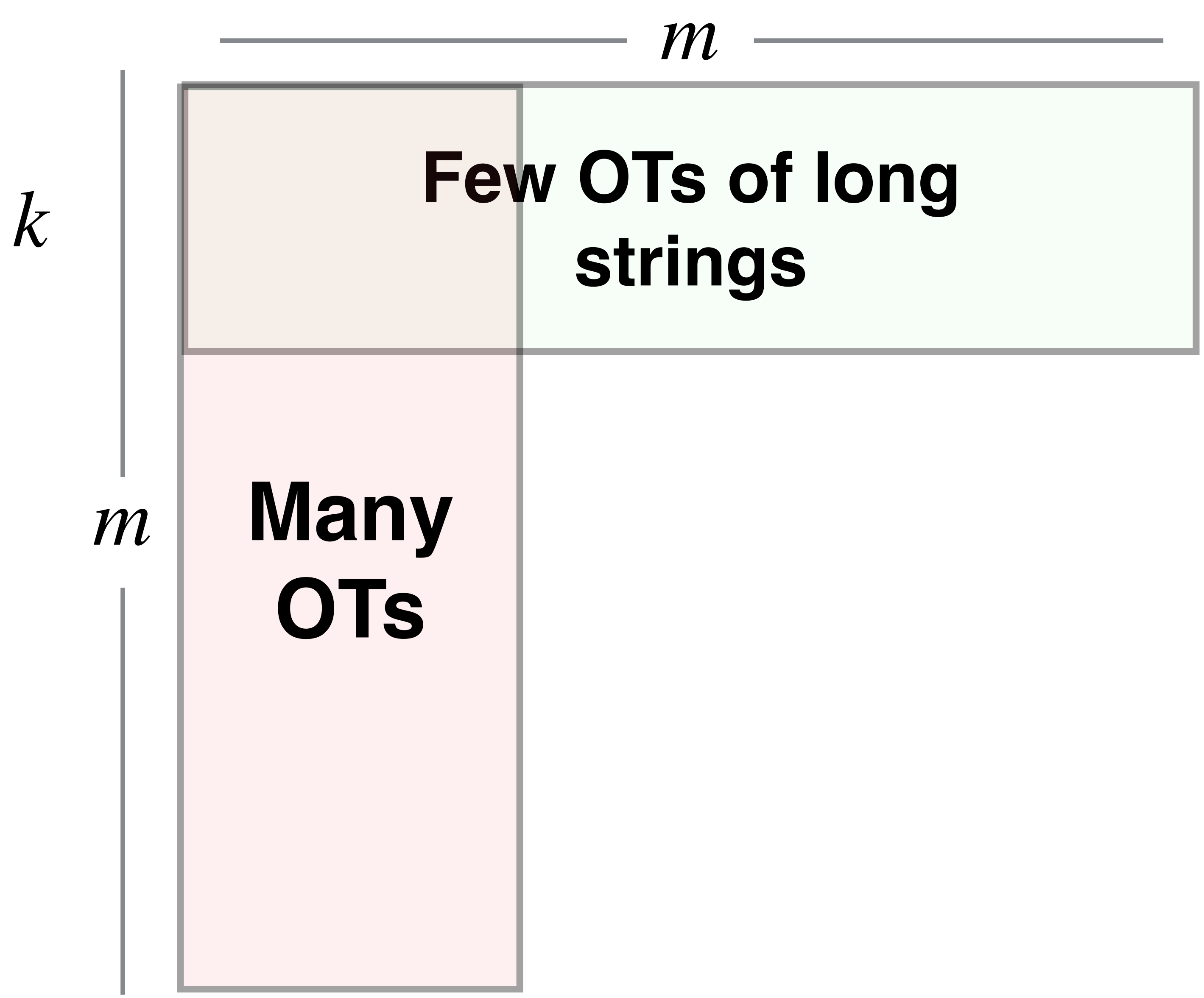
IKNP - Idea

m

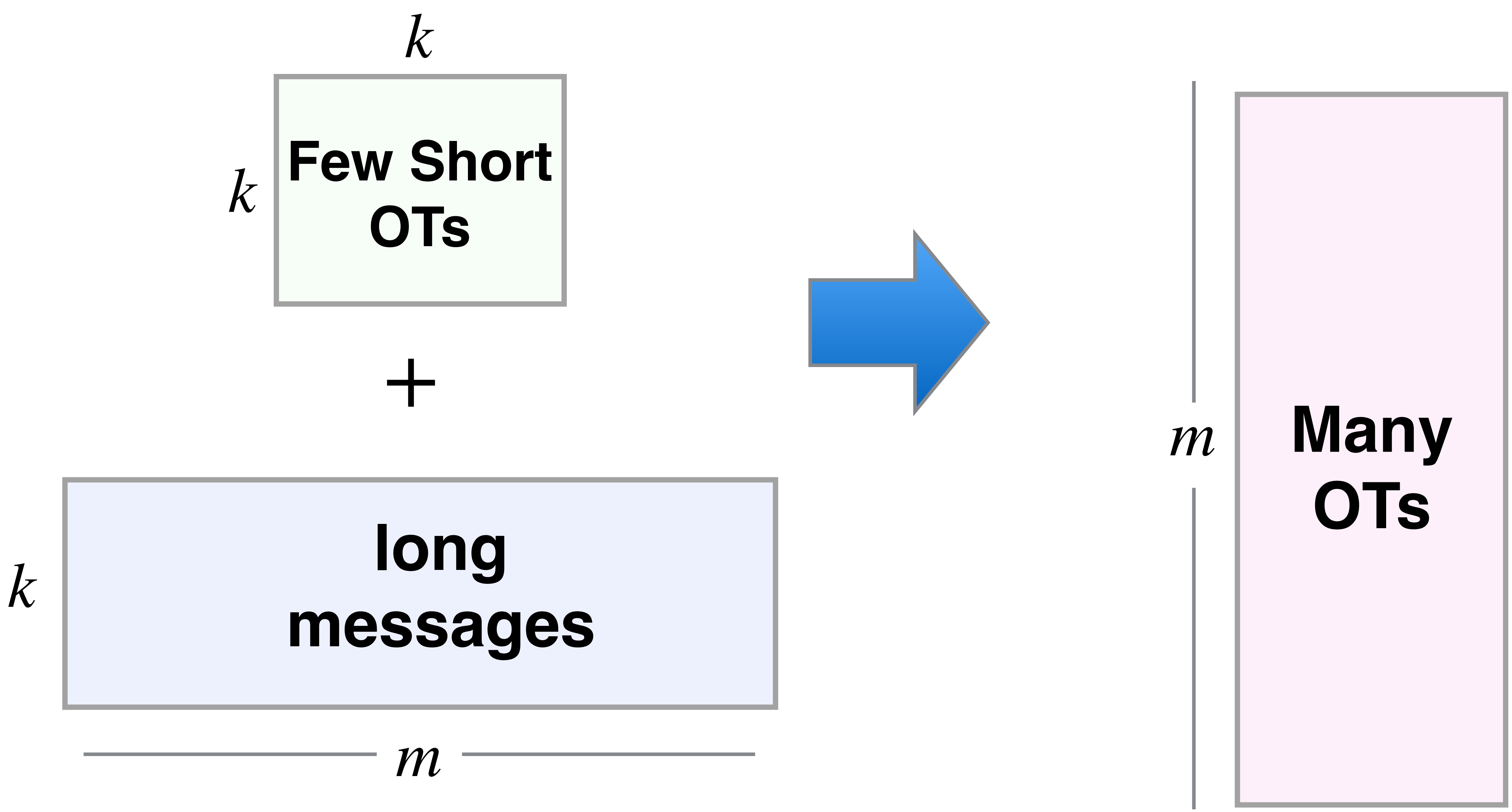
k

Few OTs of long strings

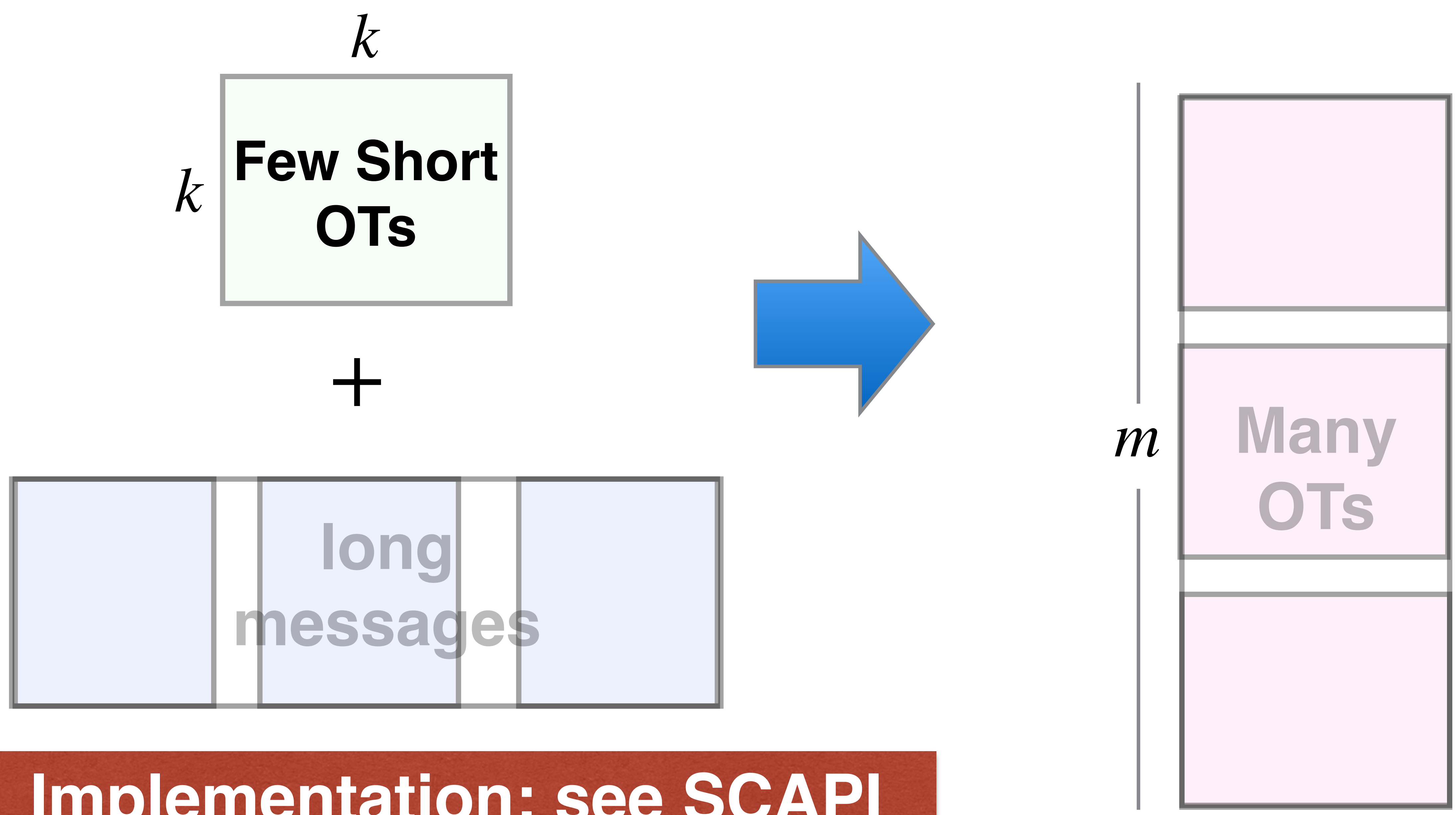
IKNP - Idea



IKNP - Implementation



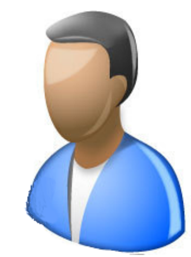
In Practice [ALSZ13]



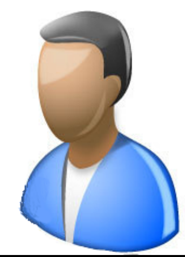
Implementation: see SCAPI

IKNP

$\{x_j^0, x_j^1\}_{j=1}^m$ 

 $\mathbf{r} = (r_1, \ldots, r_m)$

IKNP

$\{x_j^0, x_j^1\}_{j=1}^m$   $\mathbf{r} = (r_1, \ldots, r_m)$

$\mathbf{s} = (s_1, \ldots, s_\ell)$

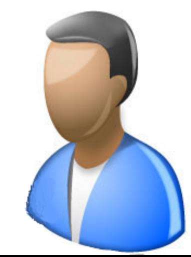
$\mathbf{k}_1^{s_1}, \ldots, \mathbf{k}_\ell^{s_\ell}$

Base OTs

$\{\mathbf{k}_i^0, \mathbf{k}_i^1\}_{i=1}^\ell$

IKNP

$\{x_j^0, x_j^1\}_{j=1}^m$



$\mathbf{r} = (r_1, \ldots, r_m)$

$\mathbf{s} = (s_1, \ldots, s_\ell)$

$\mathbf{k}_1^{s_1}, \ldots, \mathbf{k}_\ell^{s_\ell}$

Base OTs

$\{\mathbf{k}_i^0, \mathbf{k}_i^1\}_{i=1}^\ell$

$\xleftarrow{\mathbf{u}^1, \ldots, \mathbf{u}^\ell}$

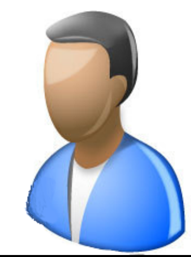
$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$ *

$\mathbf{Q}$

$\mathbf{T}$

IKNP

$\{x_j^0, x_j^1\}_{j=1}^{m}$



$\mathbf{r} = (r_1, \ldots, r_m)$

$\mathbf{s} = (s_1, \ldots, s_\ell)$

$\mathbf{k}_1^{s_1}, \ldots, \mathbf{k}_\ell^{s_\ell}$

Base OTs

$\{\mathbf{k}_i^0, \mathbf{k}_i^1\}_{i=1}^{\ell}$

$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$ *

$\xleftarrow{\mathbf{u}^1, \ldots, \mathbf{u}^\ell}$

$\mathbf{Q}$ $\mathbf{T}$

* $y_j^0 = x_j^0 \oplus H(\mathbf{q}_j)$

$y_j^1 = x_j^1 \oplus H(\mathbf{q}_j \oplus \mathbf{s})$

$\xrightarrow{y_j^0, y_j^1}$

When Moving to Malicious

When Moving to Malicious

- The protocol is already secure with respect to malicious **Sender**!

When Moving to Malicious

- The protocol is already secure with respect to malicious **Sender**!
- Malicious **Receiver** may send **inconsistent** $\mathbf{r}$ with each $\mathbf{u}^i$ message

When Moving to Malicious

- The protocol is already secure with respect to malicious **Sender**!
- Malicious **Receiver** may send <u>**inconsistent**</u> $\mathbf{r}$ with each $\mathbf{u}^i$ message
  - Learns bits of $\mathbf{s}$

When Moving to Malicious

- The protocol is already secure with respect to malicious **Sender**!
- Malicious **Receiver** may send **inconsistent** **r** with each $\mathbf{u}^i$ message
    - Learns bits of **s**

REMEMBER: if **Receiver** learns **s**, it gets ALL **Sender**'s inputs!

When Moving to Malicious

- The protocol is already secure with respect to malicious **Sender**!
- Malicious **Receiver** may send **inconsistent** **r** with each $\mathbf{u}^i$ message
  - Learns bits of **s**

REMEMBER: if **Receiver** learns **s**, it gets ALL **Sender**'s inputs!

- We add consistency check of **r**

When Moving to Malicious

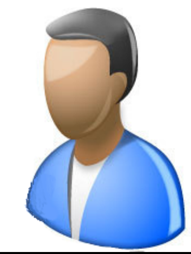
- The protocol is already secure with respect to malicious **Sender**!
- Malicious **Receiver** may send **inconsistent** $\mathbf{r}$ with each $\mathbf{u}^i$ message
  - Learns bits of $\mathbf{s}$

REMEMBER: if **Receiver** learns $\mathbf{s}$, it gets ALL **Sender**'s inputs!

- We add consistency check of $\mathbf{r}$
- **Sender** checks that **Receiver** uses the same $\mathbf{r}$ with each $\mathbf{u}^i$

The Protocol

$\{x_j^0, x_j^1\}_{j=1}^m$



$\mathbf{r} = (r_1, \ldots, r_m)$

Base OTs

$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$

$\xleftarrow{\mathbf{u}^1, \ldots, \mathbf{u}^\ell}$

Q

T

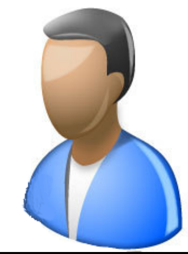
$y_j^0 = x_j^0 \oplus H(\mathbf{q}_j)$

$y_j^1 = x_j^1 \oplus H(\mathbf{q}_j \oplus \mathbf{s})$

$\xrightarrow{y_j^0, y_j^1}$

The Protocol

$\{x_j^0, x_j^1\}_{j=1}^m$



$\mathbf{r} = (r_1, \ldots, r_m)$

Base OTs

$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$

$\xleftarrow{\mathbf{u}^1, \ldots, \mathbf{u}^\ell}$

Q

T

Consistency Check of **r**

$y_j^0 = x_j^0 \oplus H(\mathbf{q}_j)$

$y_j^1 = x_j^1 \oplus H(\mathbf{q}_j \oplus \mathbf{s})$

$\xrightarrow{y_j^0, y_j^1}$

The Consistency Checks

Consistency Check

$$\mathbf{u}^{i} = G(\mathbf{k}_{i}^{0}) \oplus G(\mathbf{k}_{i}^{1}) \oplus \mathbf{r}$$

$$\mathbf{u}^{j} = G(\mathbf{k}_{j}^{0}) \oplus G(\mathbf{k}_{j}^{1}) \oplus \mathbf{r}$$

Consistency Check

$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$

$\mathbf{u}^j = G(\mathbf{k}_j^0) \oplus G(\mathbf{k}_j^1) \oplus \mathbf{r}$

$$\mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r}$$

$$\mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r}$$

Consistency Check

$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$

$\mathbf{u}^j = G(\mathbf{k}_j^0) \oplus G(\mathbf{k}_j^1) \oplus \mathbf{r}$

$$\oplus \begin{array}{l} \mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r} \\ \mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r} \\ \hline \end{array}$$

Consistency Check

$$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$$

$$\mathbf{u}^j = G(\mathbf{k}_j^0) \oplus G(\mathbf{k}_j^1) \oplus \mathbf{r}$$

$$\oplus \begin{array}{l} \mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r} \\ \mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r} \end{array}$$

$$\mathbf{u}^i \oplus \mathbf{u}^j = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{t}_j^0 \oplus \mathbf{t}_j^1$$

Consistency Check

$$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$$
$$\mathbf{u}^j = G(\mathbf{k}_j^0) \oplus G(\mathbf{k}_j^1) \oplus \mathbf{r}$$

$$\oplus \begin{array}{l} \mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r} \\ \mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r} \end{array}$$

$$\mathbf{u}^i \oplus \mathbf{u}^j = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{t}_j^0 \oplus \mathbf{t}_j^1$$

$$\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j} \; ?= \; \mathbf{t}_i^{1-s_i} \oplus \mathbf{t}_j^{1-s_j}$$

Consistency Check

$$\mathbf{u}^i = G(\mathbf{k}_i^0) \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$$
$$\mathbf{u}^j = G(\mathbf{k}_j^0) \oplus G(\mathbf{k}_j^1) \oplus \mathbf{r}$$

$$\oplus \begin{array}{l} \mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r} \\ \mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r} \end{array}$$

$$\mathbf{u}^i \oplus \mathbf{u}^j = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{t}_j^0 \oplus \mathbf{t}_j^1$$

$$\boxed{\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j}} \;?= \boxed{\mathbf{t}_i^{1-s_i} \oplus \mathbf{t}_j^{1-s_j}}$$

$$\mathrm{H}(\boxed{\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j}}) \;?= \; \mathrm{H}(\boxed{\mathbf{t}_i^{1-s_i} \oplus \mathbf{t}_j^{1-s_j}})$$

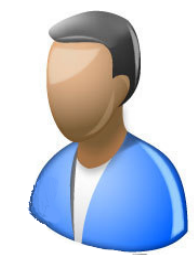
Consistency Check

$$h_{i,j}^{0,0} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{0,1} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^1)$$

$$h_{i,j}^{1,0} = H(\mathbf{t}_i^1 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{1,1} = H(\mathbf{t}_i^1 \oplus \mathbf{t}_j^1)$$



For every pair (i,j)

Consistency Check

$$h_{i,j}^{0,0} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{0,1} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^1)$$

$$h_{i,j}^{1,0} = H(\mathbf{t}_i^1 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{1,1} = H(\mathbf{t}_i^1 \oplus \mathbf{t}_j^1)$$

For every pair (i,j)

$$\mathbf{u}^1,\ldots,\mathbf{u}^\ell \quad \{h_{i,j}^{0,0}, h_{i,j}^{0,1}, h_{i,j}^{1,0}, h_{i,j}^{1,1}\}_{i,j}$$

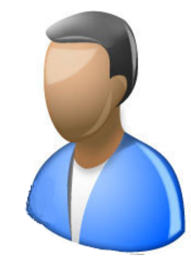
Consistency Check

$$h_{i,j}^{0,0} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{0,1} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^1)$$

$$h_{i,j}^{1,0} = H(\mathbf{t}_i^1 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{1,1} = H(\mathbf{t}_i^1 \oplus \mathbf{t}_j^1)$$



For every pair (i,j)

$$\mathbf{u}^1, \ldots, \mathbf{u}^\ell \quad \{h_{i,j}^{0,0}, h_{i,j}^{0,1}, h_{i,j}^{1,0}, h_{i,j}^{1,1}\}_{i,j}$$

Alice checks that every pair *(i,j)*:

$$h_{i,j}^{1-s_i,1-s_j} \ ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

$$h_{i,j}^{s_i,s_j} \ ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- **Our goal**: in case $\mathbf{r}^{i} \neq \mathbf{r}^{j}$, catch the adversary

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- **Our goal**: in case $\mathbf{r}^i \neq \mathbf{r}^j$, catch the adversary

$$\mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r}^i$$
$$\mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r}^j$$

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- **Our goal**: in case $\mathbf{r}^i \neq \mathbf{r}^j$, catch the adversary

$$\mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r}^i$$
$$\mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r}^j$$

- But **Receiver** sends $(h_{i,j}^{0,0}, h_{i,j}^{0,1}, h_{i,j}^{1,0}, h_{i,j}^{1,1})$ such that:

$$h^{0,0} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^0) \quad h^{1,1} = H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_j^1 \oplus \mathbf{t}_j^1)$$
$$h^{0,1} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^1) \quad h^{1,0} = H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^1 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- **Our goal**: in case $\mathbf{r}^i \neq \mathbf{r}^j$, catch the adversary

$$\mathbf{u}^i = \mathbf{t}_i^0 \oplus \mathbf{t}_i^1 \oplus \mathbf{r}^i$$
$$\mathbf{u}^j = \mathbf{t}_j^0 \oplus \mathbf{t}_j^1 \oplus \mathbf{r}^j$$

- But **Receiver** sends $(h_{i,j}^{0,0}, h_{i,j}^{0,1}, h_{i,j}^{1,0}, h_{i,j}^{1,1})$ such that:

$$h^{0,0} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^0) \quad h^{1,1} = H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_j^1 \oplus \mathbf{t}_j^1)$$
$$h^{0,1} = H(\mathbf{t}_i^0 \oplus \mathbf{t}_j^1) \quad h^{1,0} = H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^1 \oplus \mathbf{t}_j^0)$$

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

if $\mathbf{s}_i$=0 **Passes**

if $\mathbf{s}_i$=1 **Gets caught**

Does it really work?

Alice checks that every pair *(i,j)*:

$$h_{i,j}^{1-s_i,1-s_j} \; ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

$$h_{i,j}^{s_i,s_j} \; ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

Alice checks that every pair *(i,j)*:

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- If $\mathbf{r}^i \neq \mathbf{r}^j$ then:

Alice checks that every pair *(i,j)*:

$$h_{i,j}^{1-s_i,1-s_j} ?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} ?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- If $\mathbf{r}^i \neq \mathbf{r}^j$ then:
  If the verification **passes** for (s^i,s^j) -
  the verification **fails** for $(1\text{-}s^i, 1\text{-}s^j)$

Alice checks that every pair *(i,j)*:

$$h_{i,j}^{1-s_i,1-s_j} \mathrel{?=} H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} \mathrel{?=} H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

Does it really work?

- If $\mathbf{r}^i \neq \mathbf{r}^j$ then:
  If the verification **passes** for (s^i, s^j) -
  the verification **fails** for $(1\text{-}s^i, 1\text{-}s^j)$

- It can succeed only with 2-out-of-4 possibilities of (s^i, s^j)
  With probability 1/2, we catch the adversary!

Alice checks that every pair *(i,j)*:

$$h_{i,j}^{1-s_i,1-s_j} \;?= H(\mathbf{u}^i \oplus \mathbf{u}^j \oplus \mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$
$$h_{i,j}^{s_i,s_j} \;?= H(\mathbf{t}_i^{s_i} \oplus \mathbf{t}_j^{s_j})$$

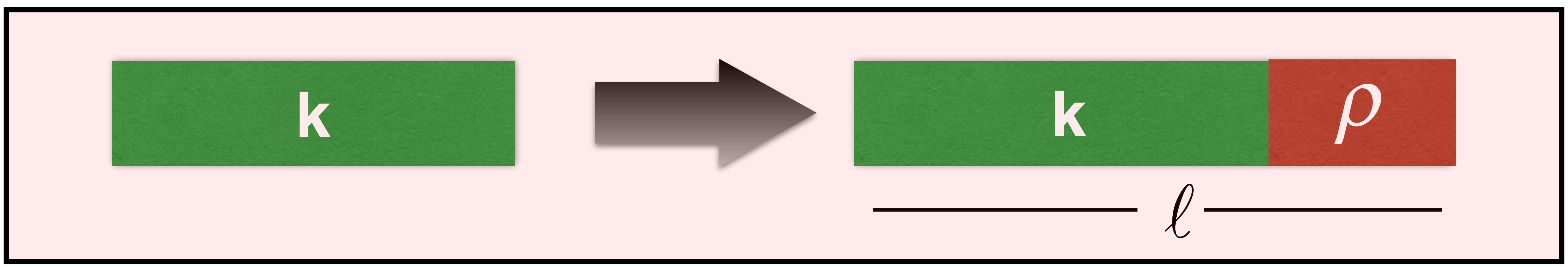
Consistency Check

Consistency Check

- **Bob** can still learn *t* bits of **s**, with probability 2^{-t}
  - By guessing s_i, can pass verification of (i,j) for all j

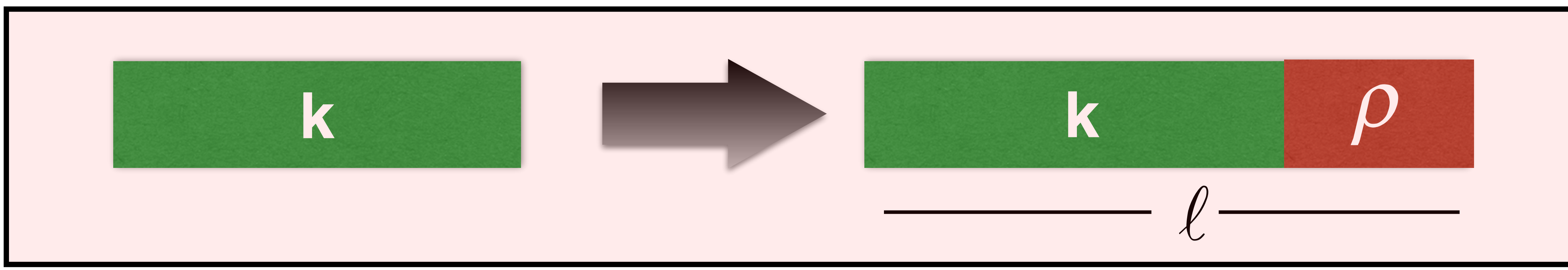
Consistency Check

- **Bob** can still learn *t* bits of **s**, with probability 2^{-t}
  - By guessing s_i, can pass verification of (i,j) for all j
- **Solution** - increase the size of **s**



Consistency Check

- **Bob** can still learn *t* bits of **s**, with probability 2^{-t}
  - By guessing s_i, can pass verification of (i,j) for all j
- **Solution** - increase the size of **s**



- With probability $1-2^{-\rho}$, still k bits of **s** are completely hidden!

$$y_j^0 = x_j^0 \oplus H(\mathbf{q}_j)$$

$$y_j^1 = x_j^1 \oplus H(\mathbf{q}_j \oplus \mathbf{s})$$

Some concrete numbers…

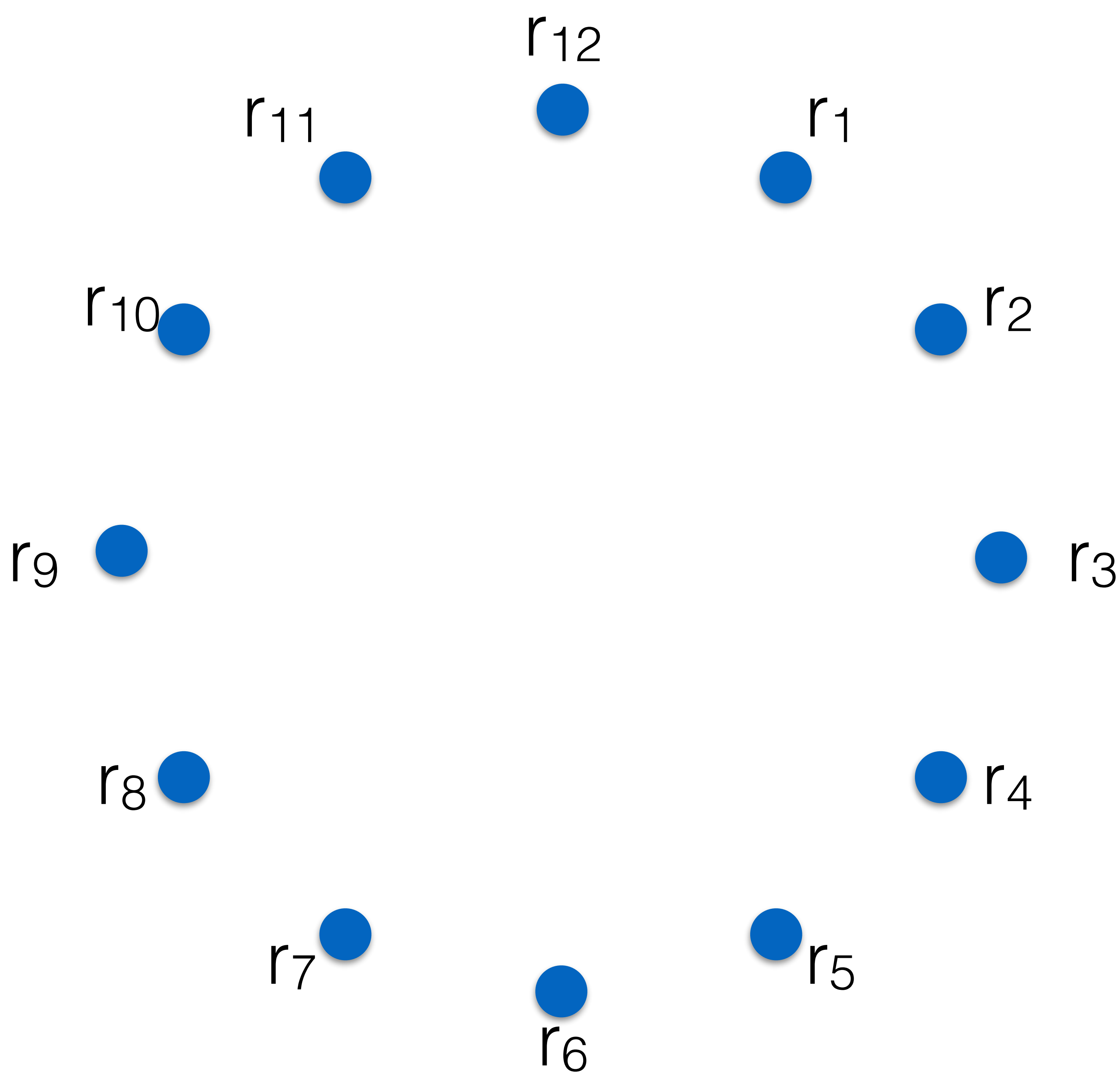
Some concrete numbers…

- Typical security parameter: 128
- Typical statistical sec. parameter: 40
- Overall number of base OTs: 168
  (**Reminder**: [NNOB12] uses 8/3k = 341 base OTs)

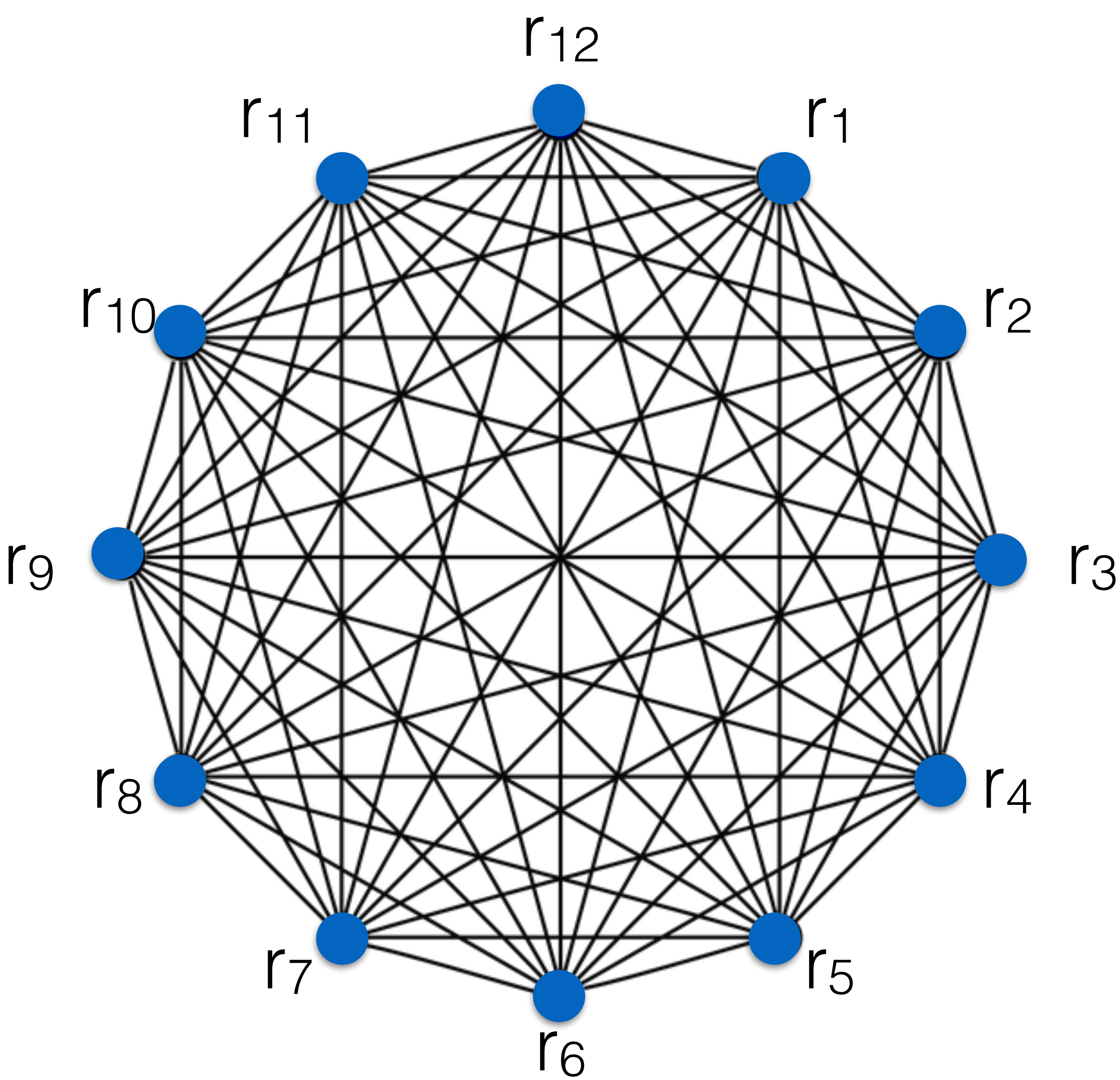
Some concrete numbers…

- Typical security parameter: 128
- Typical statistical sec. parameter: 40
- Overall number of base OTs: 168
(**Reminder**: [NNOB12] uses 8/3k = 341 base OTs)
- Checks: all pairs ~ 14028
- We have to reduce the number of checks!
(at the expense of increasing the number of base-OTs)

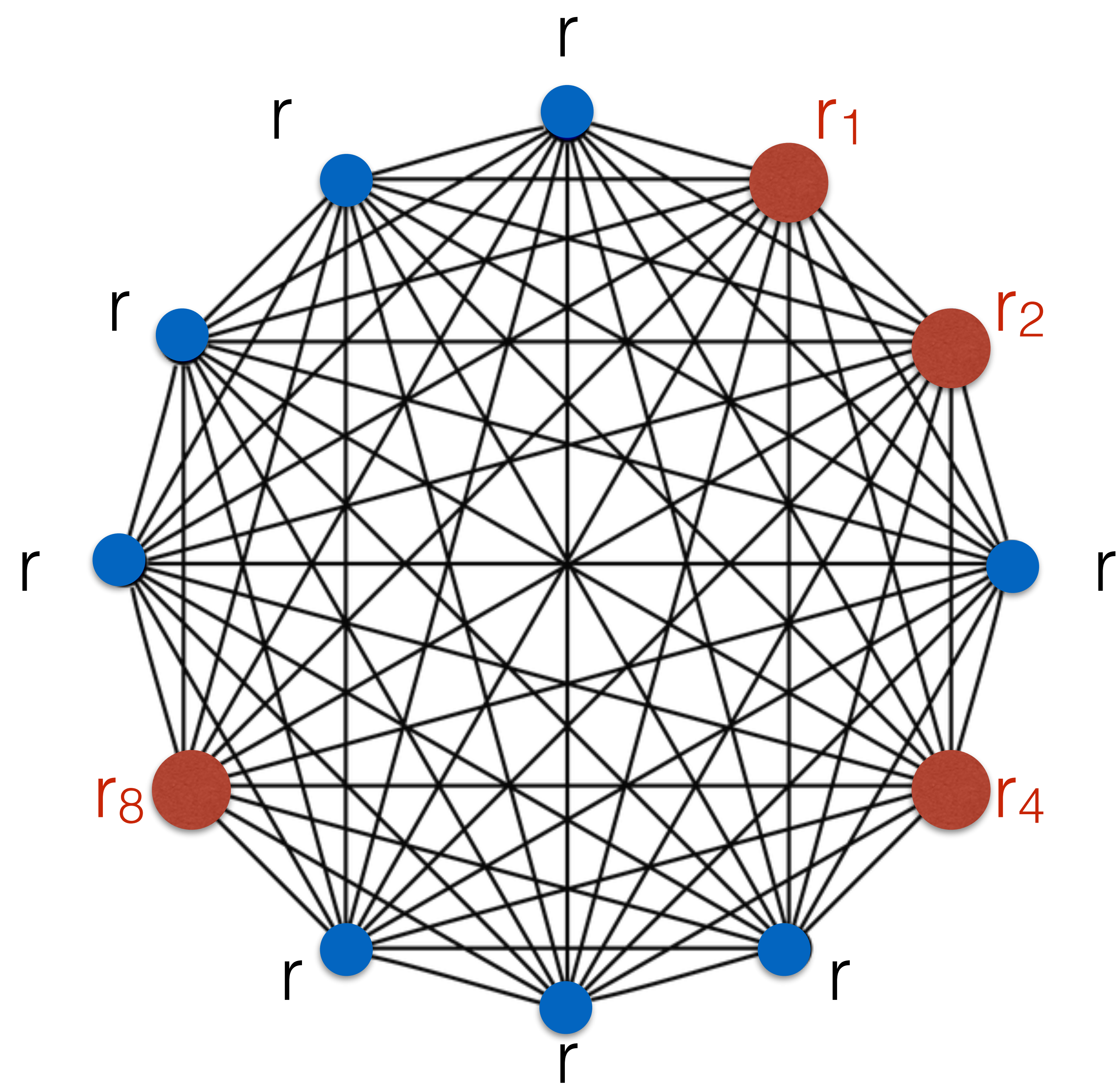
How many checks do we really need?



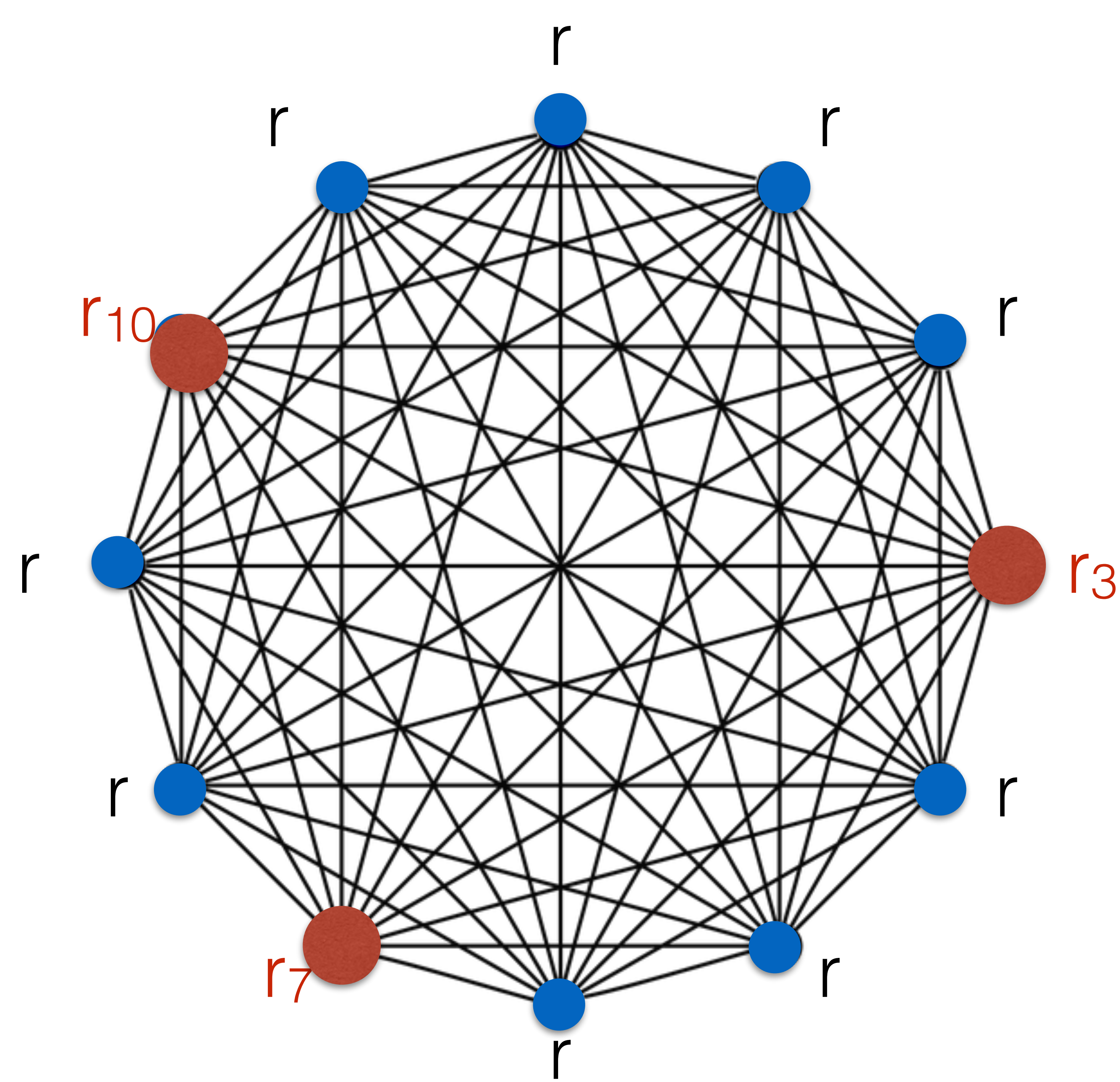
How many checks do we really need?



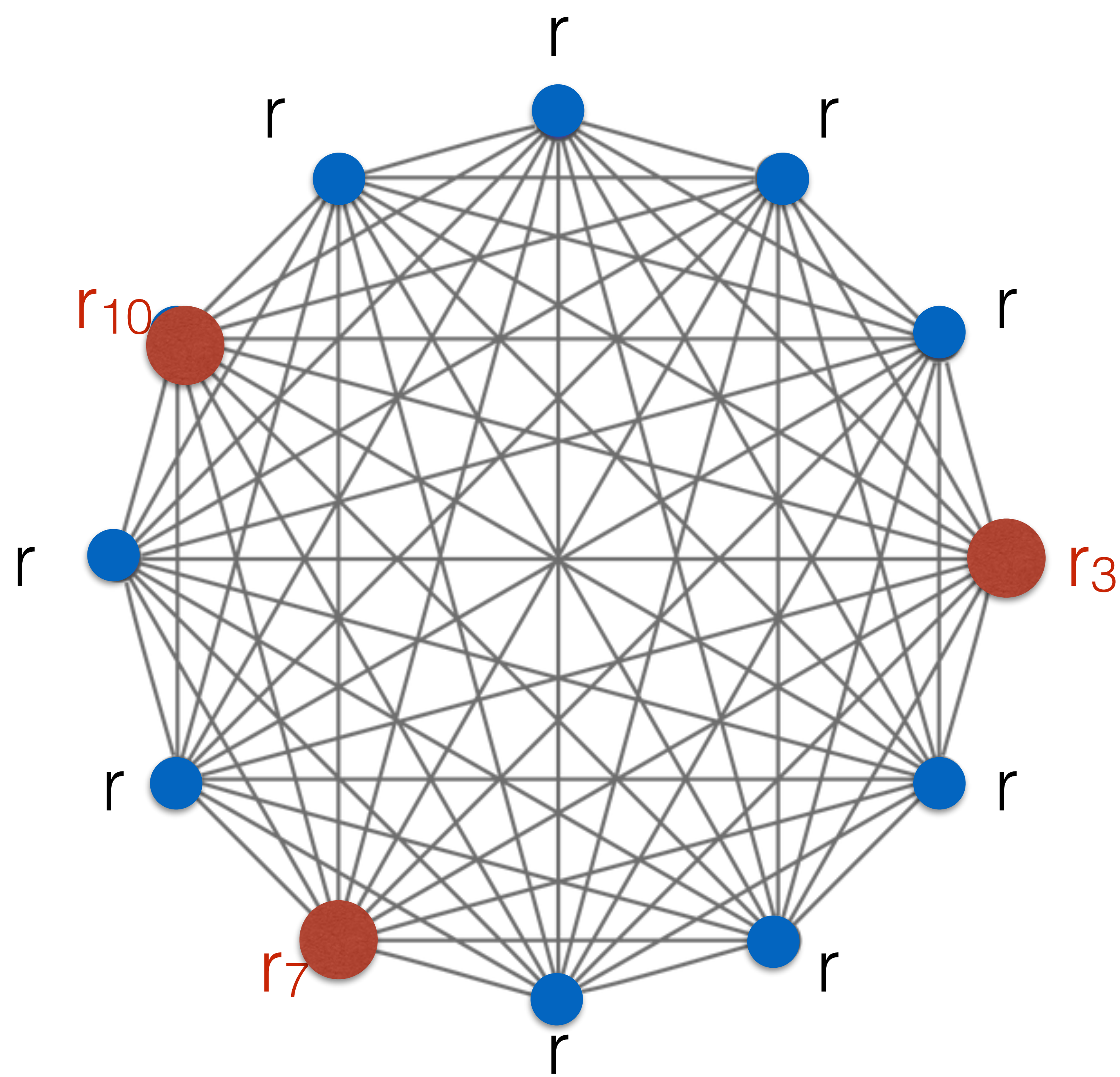
How many checks do we really need?



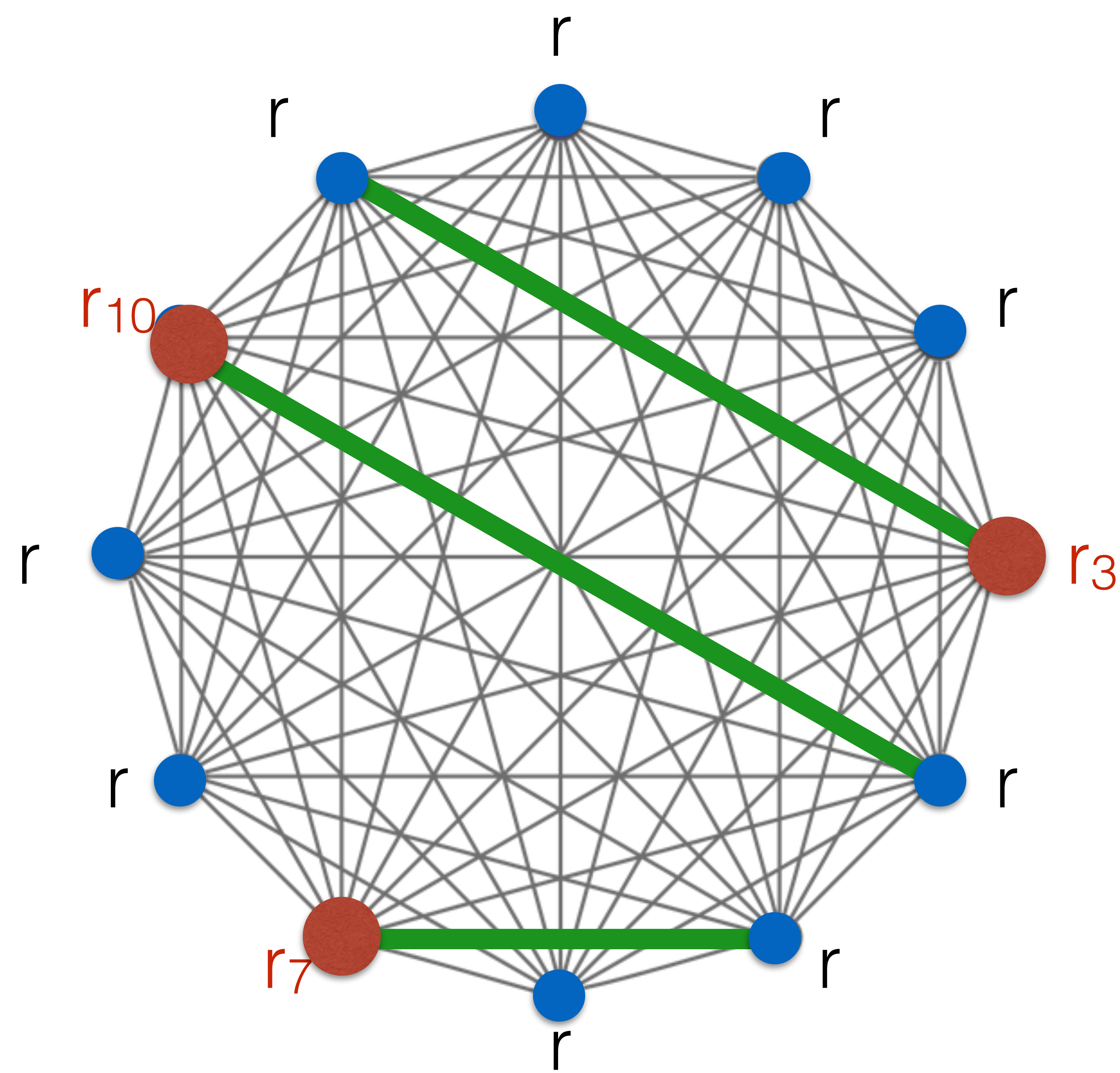
How many checks do we really need?



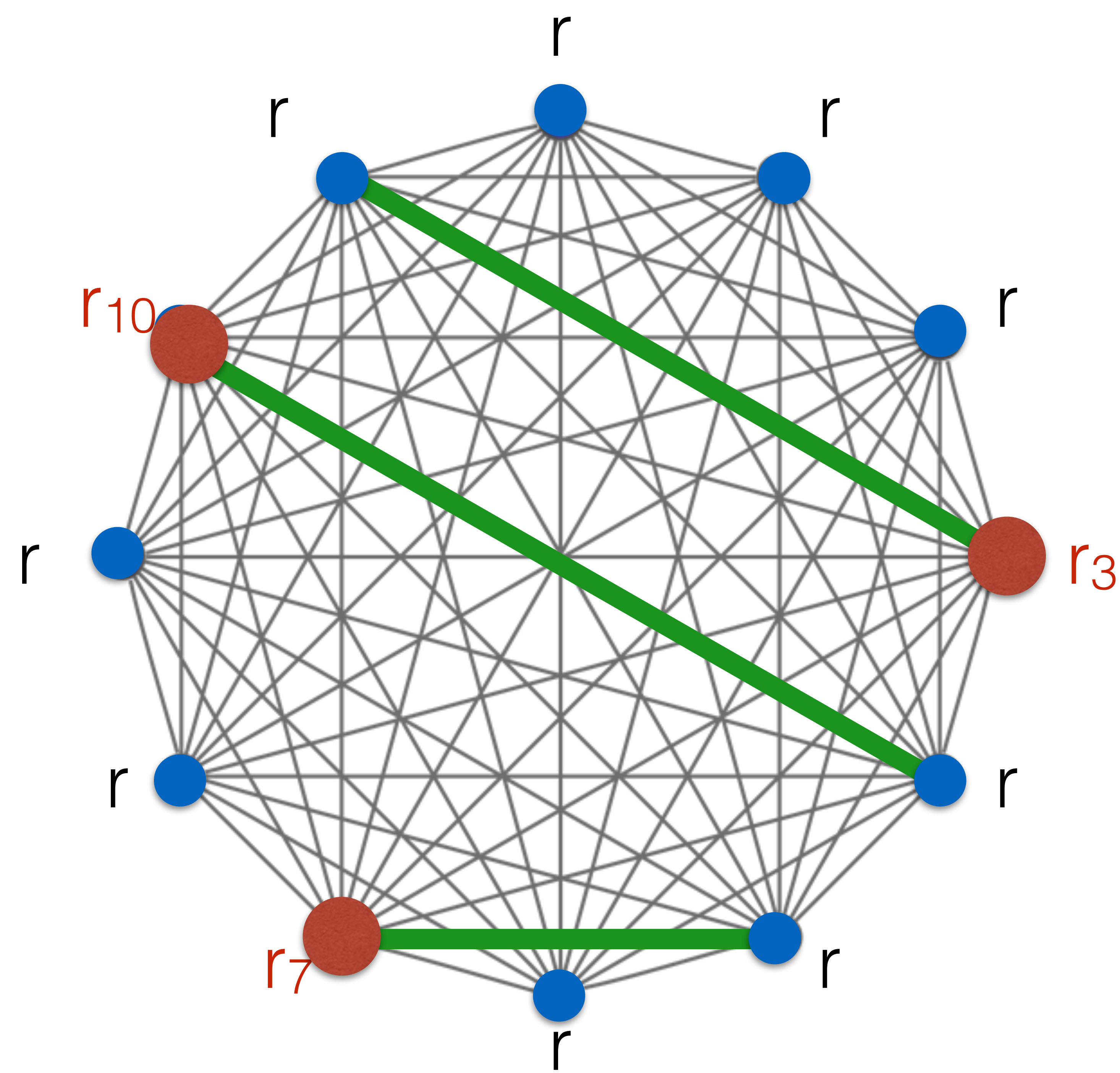
How many checks do we really need?



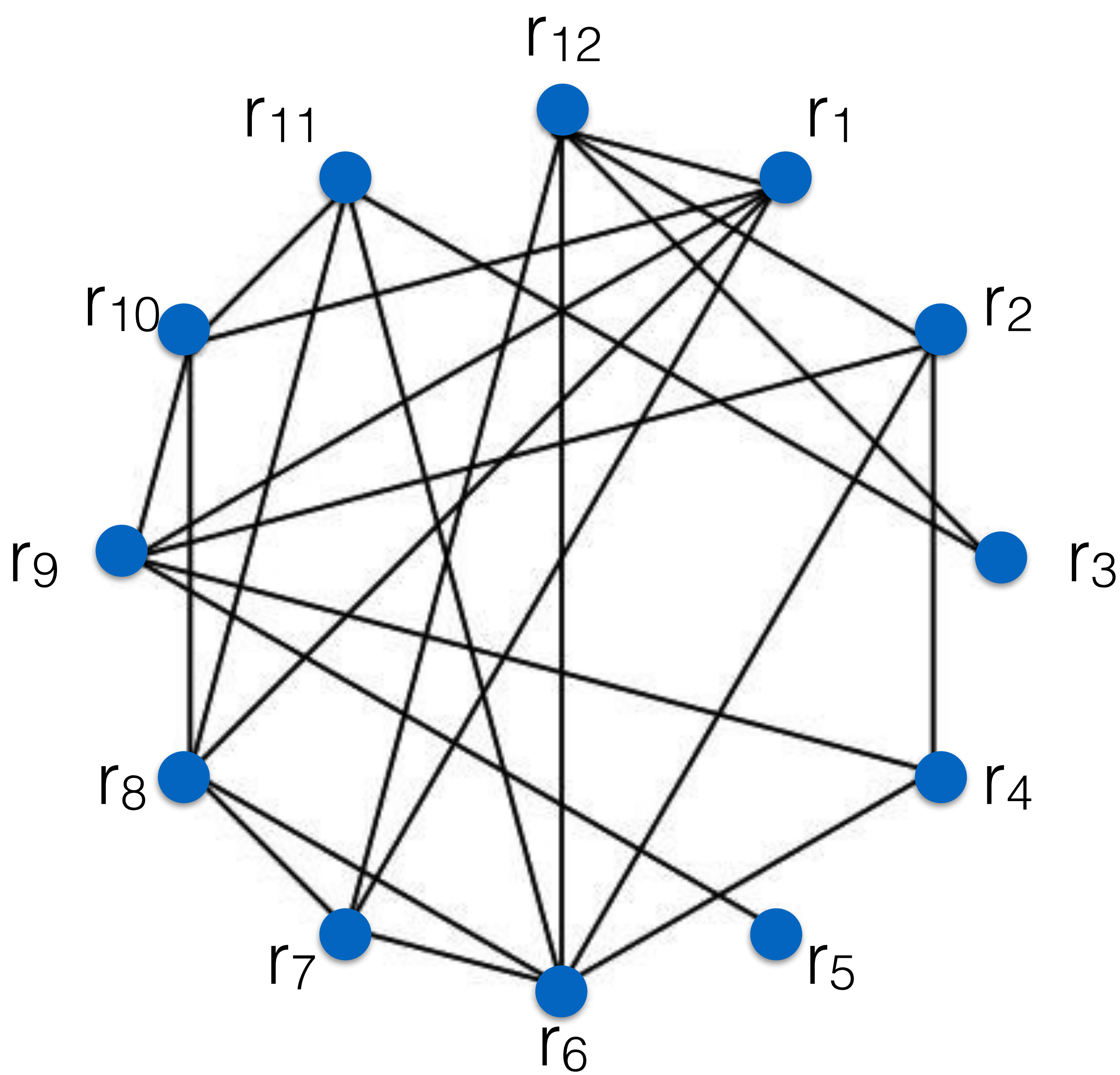
How many checks do we really need?



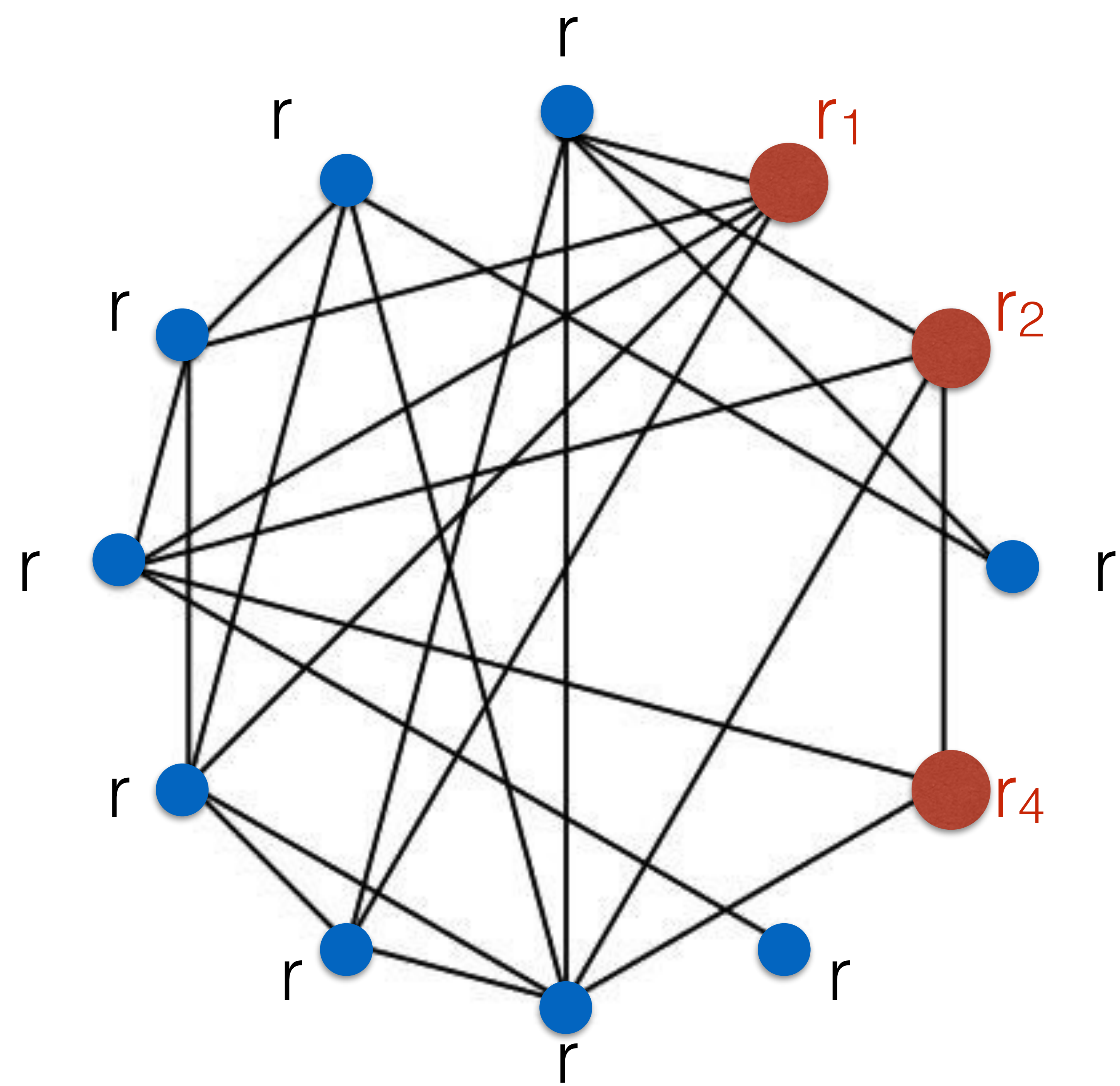
The needed property:
For any “large enough" set of **bad** vertices
(> p=40), **there exists** p-matching with the **good** vertices



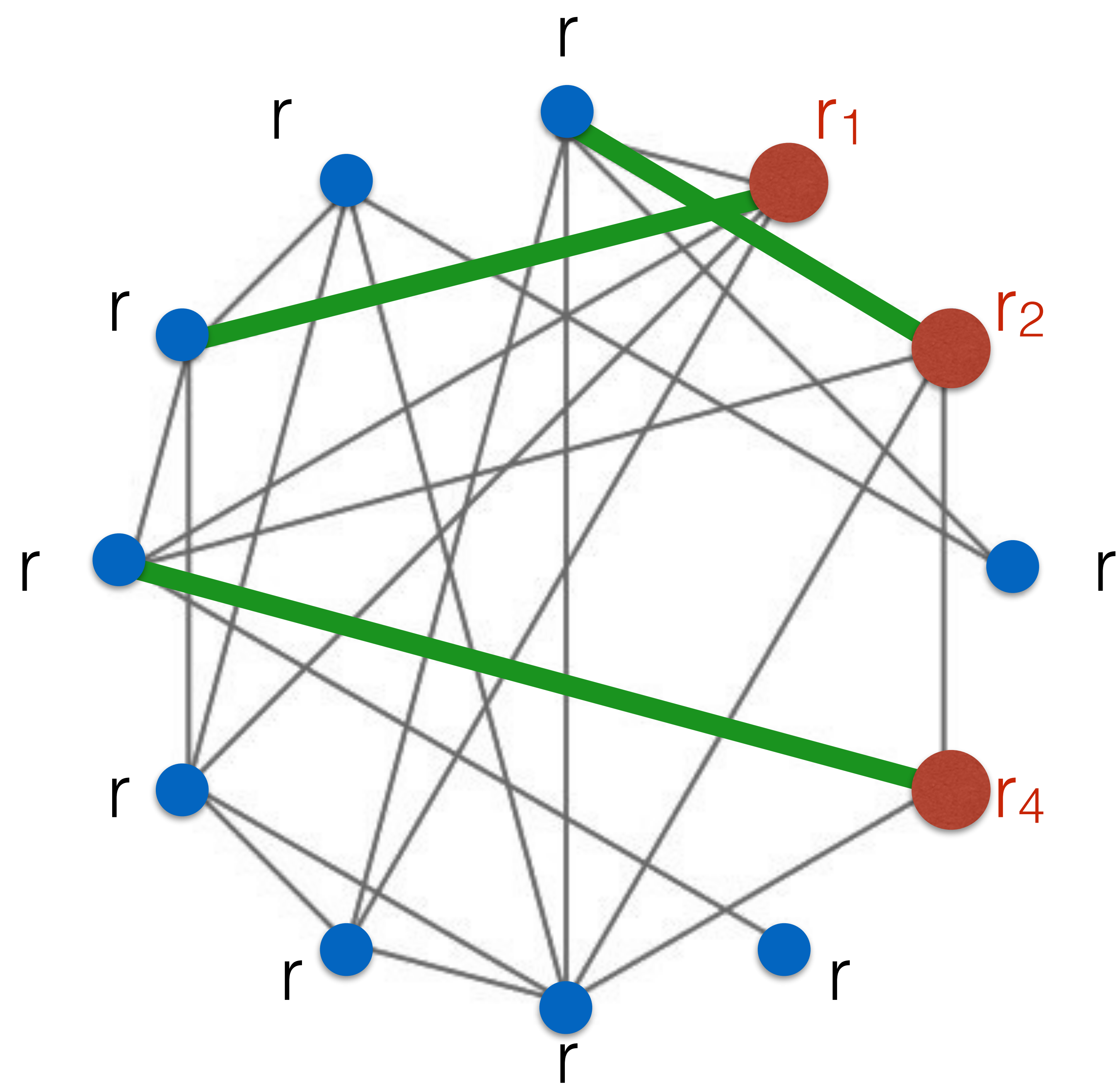
How many checks do we really need?



How many checks do we really need?



How many checks do we really need?



How Many Checks?

The needed property:
For any “large enough" set of **bad** vertices
(> p=40), **there exists** p-matching with the **good** vertices

How Many Checks?

The needed property:
For any “large enough" set of **bad** vertices
(> p=40), **there exists** p-matching with the **good** vertices

- We show that random **d-regular graph** satisfies the above (for appropriate set of parameters)

How Many Checks?

The needed property:
For any “large enough" set of **bad** vertices
(> p=40), **there exists** p-matching with the **good** vertices

- We show that random **d-regular graph** satisfies the above (for appropriate set of parameters)
  - For k=128, p=40
    - 168 base OTs, complete graph: 14028
    - 190 base OTs, d=2, checks: 380
    - 177 base OTs, d=3, checks: 531

How Many Checks?

The needed property:
For any “large enough" set of **bad** vertices ($> p=40$), **there exists** p-matching with the **good** vertices

- We show that random **d-regular graph** satisfies the above (for appropriate set of parameters)
  - For k=128, p=40
    - 168 base OTs, complete graph: 14028
    - 190 base OTs, d=2, checks: 380
    - 177 base OTs, d=3, checks: 531
- **Covert**: probability 1/2, just random 7 checks!

Instantiation of **H**

Correlation Robustness:

$$\{H(t_1 \oplus \mathbf{s}),\ldots,H(t_\ell \oplus \mathbf{s})\} \stackrel{c}{=} U_{\ell \times n}$$

$$\mathbf{s} \leftarrow_R \{0,1\}^k$$

Instantiation of **H**

Correlation Robustness:

$$\{H(t_1 \oplus \mathbf{s}), \ldots, H(t_\ell \oplus \mathbf{s})\} \stackrel{c}{=} U_{\ell \times n}$$

$$\mathbf{s} \leftarrow_R \{0,1\}^k$$

k-Min Entropy Correlation Robustness:

$$\{H(t_1 \oplus \mathbf{s}), \ldots, H(t_\ell \oplus \mathbf{s})\} \stackrel{c}{=} U_{\ell \times n}$$

$\mathbf{s}$ is taken from a source χ with min entropy k

Performance

Empirical Evaluation

Empirical Evaluation

- Benchmark: 2^{23}=8M OTs

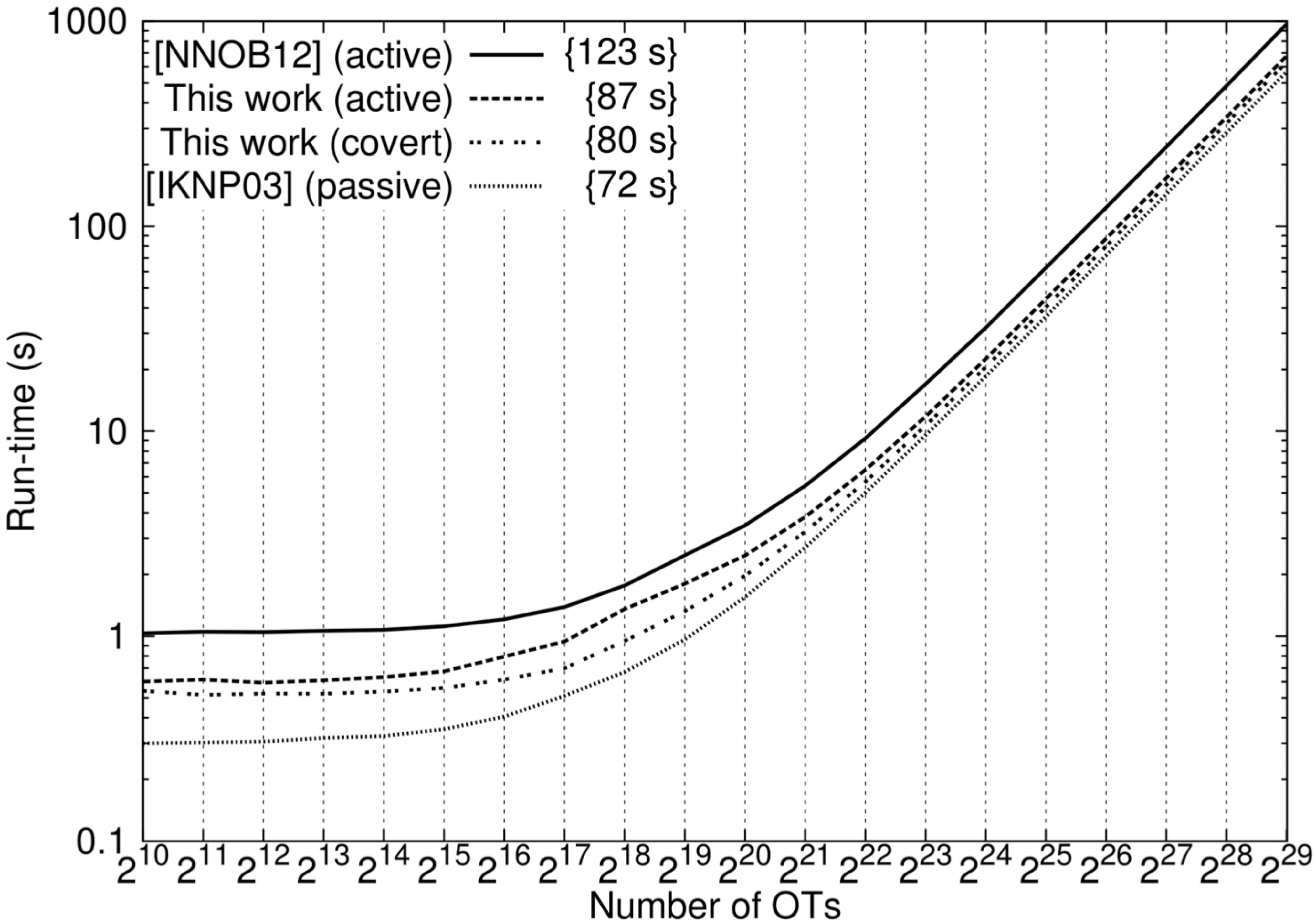
Empirical Evaluation

- Benchmark: 2^{23}=8M OTs
- **Local scenario (LAN):**
  Two servers in the same room
  (network with low latency and high bandwidth)
  12 sec (190 base OTs, 380 checks)

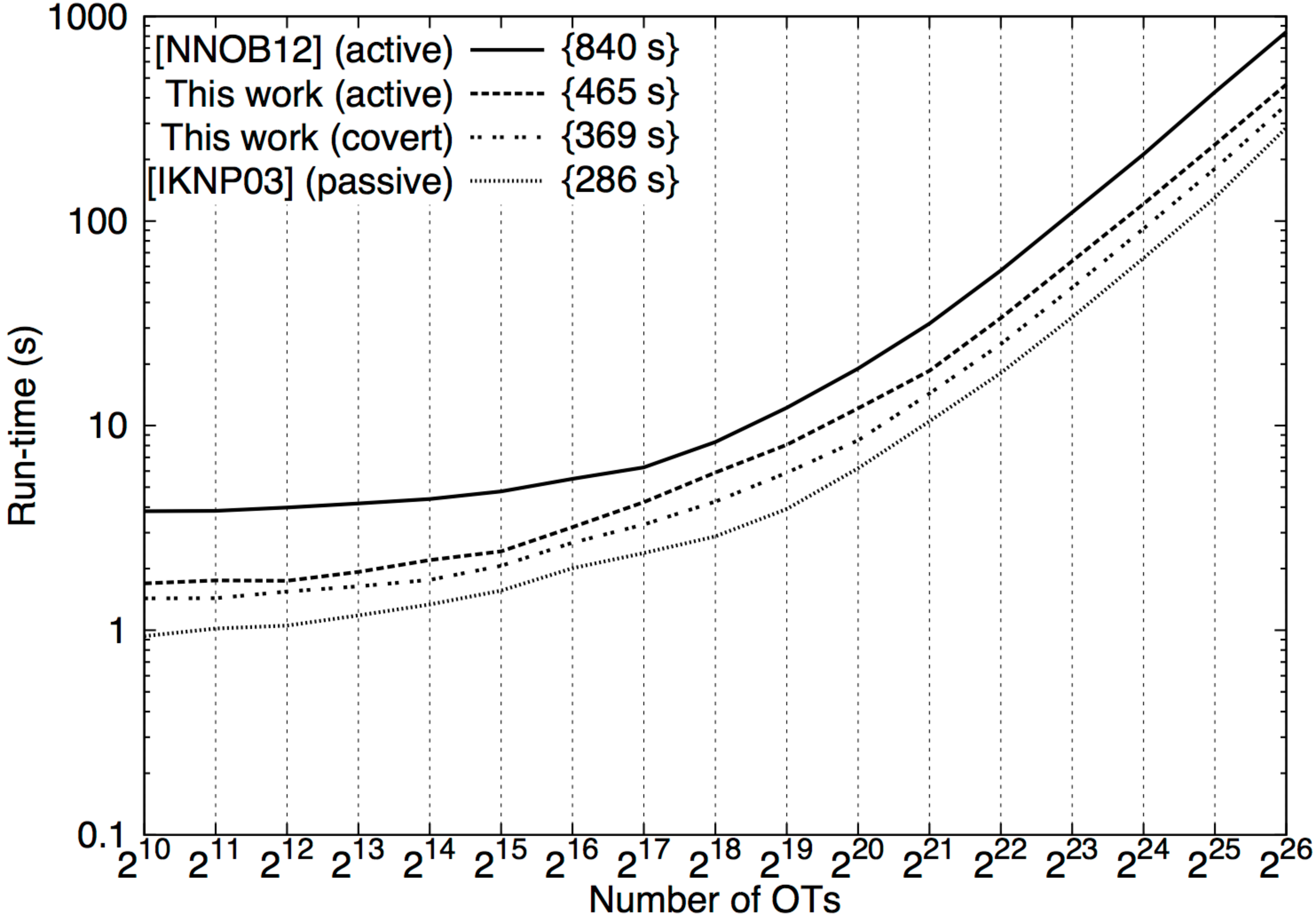
Empirical Evaluation

- Benchmark: 2^{23}=8M OTs
- **Local scenario (LAN):**
  Two servers in the same room
  (network with low latency and high bandwidth)
  12 sec (190 base OTs, 380 checks)
- **Cloud scenario (WAN):**
  Two servers in different continents
  (network with high latency and low bandwidth)
  64 sec (174 base OTs, 696 checks)

Comparison - LAN Setting



Comparison - WAN setting



Conclusions

- More efficient OT extension - more efficient protocols for MPC
- The most efficient OT extension protocol, yet
- Combination of theory and practice

Conclusions

- More efficient OT extension - more efficient protocols for MPC
- The most efficient OT extension protocol, yet
- Combination of theory and practice

Thank You!