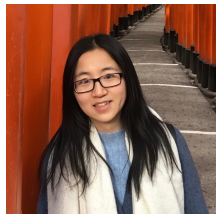
Private Set Intersection in the Internet Setting From Lightweight Oblivious PRF



Melissa Chase

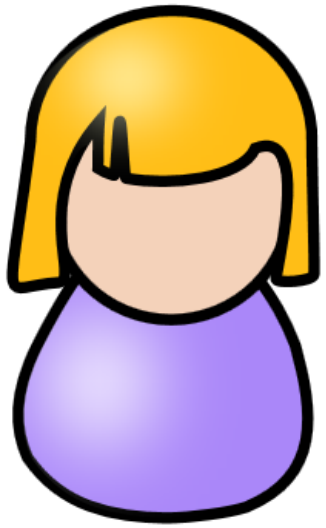
Microsoft Research

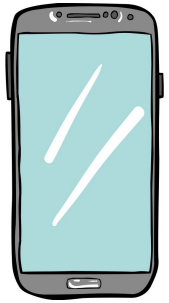
Peihan Miao

VISA Research

UIC

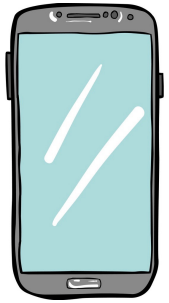
What is Private Set Intersection?



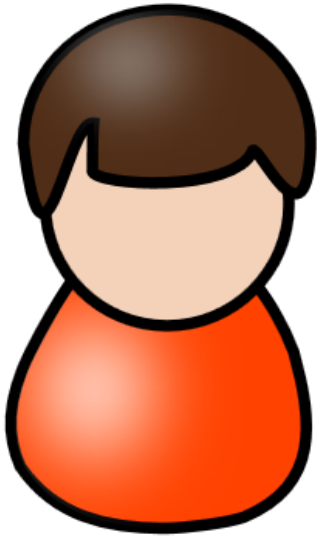


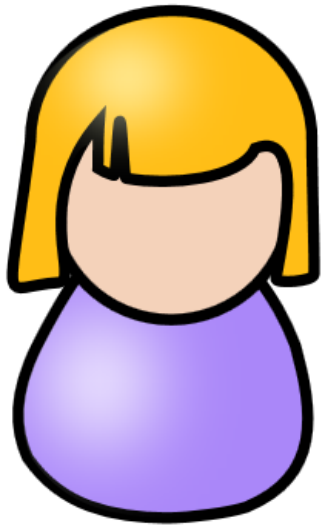
Charlie
Peihan
David
Eve
Melissa

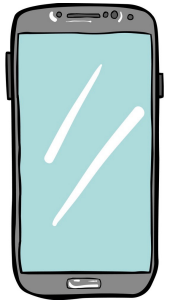
Common friends?



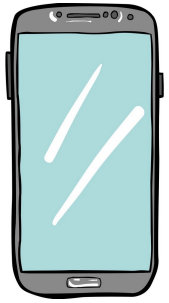
Eric
Frank
Melissa
Peihan
Grace



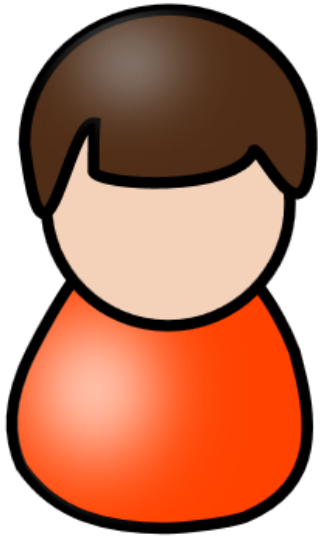


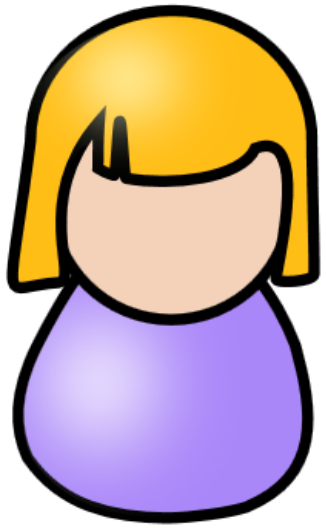


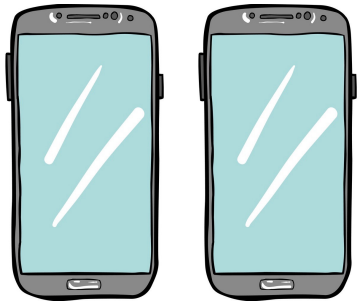
Charlie
Peihan
David
Eve
Melissa



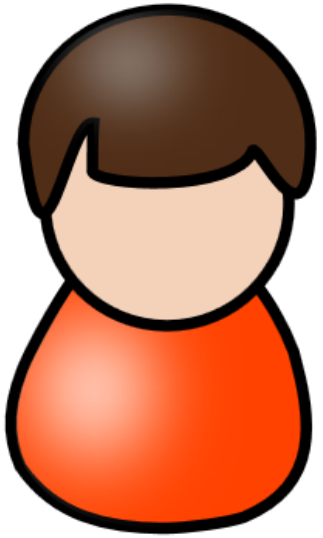
Eric
Frank
Melissa
Peihan
Grace

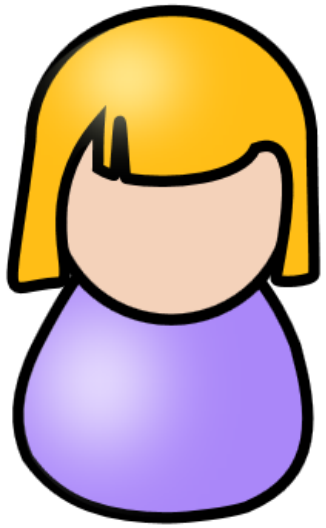


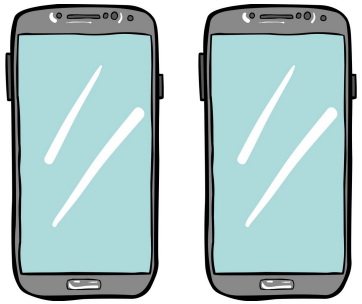




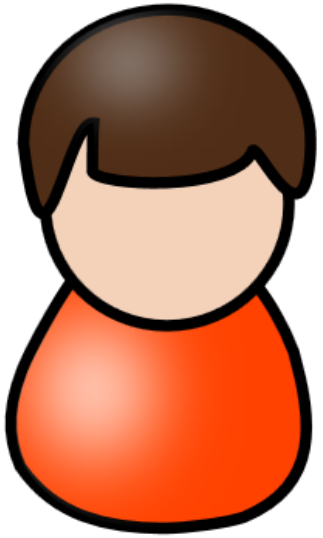
Charlie	Eric
Peihan	Frank
David	Melissa
Eve	Peihan
Melissa	Grace

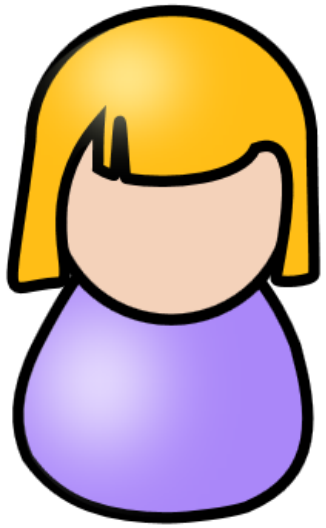


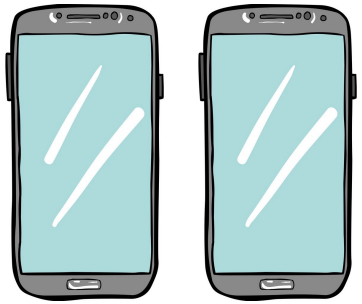




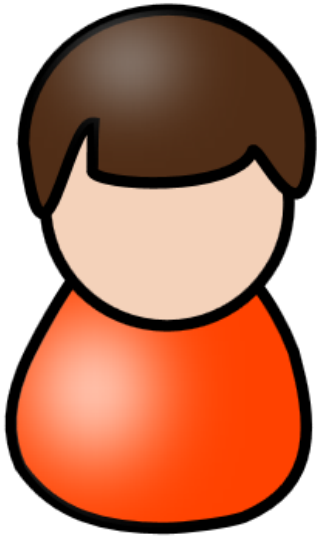
Charlie	Eric
Peihan	Frank
David	Melissa
Eve	Peihan
Melissa	Grace

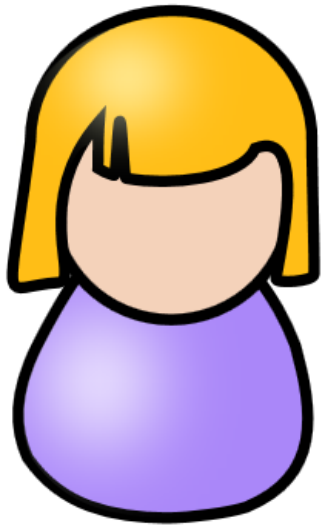


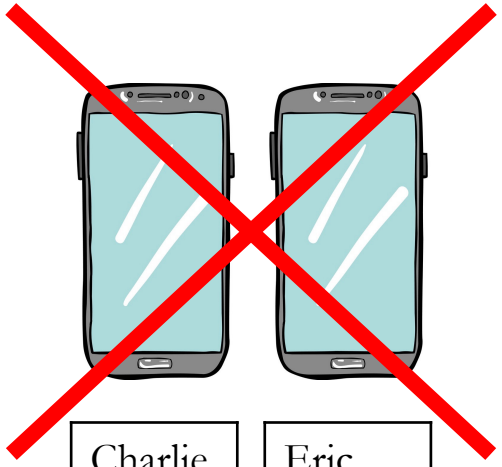




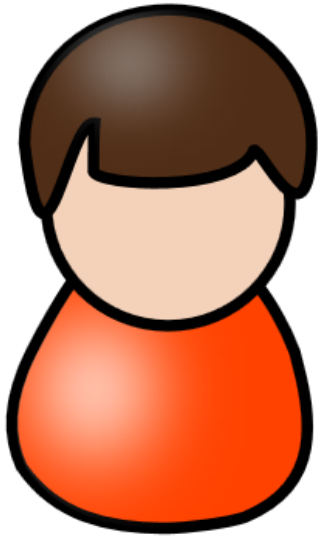
Charlie	Eric
Peihan	Frank
David	Melissa
Eve	Peihan
Melissa	Grace

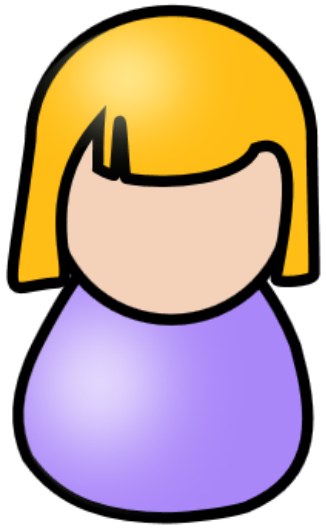


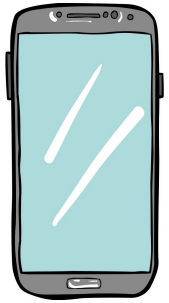




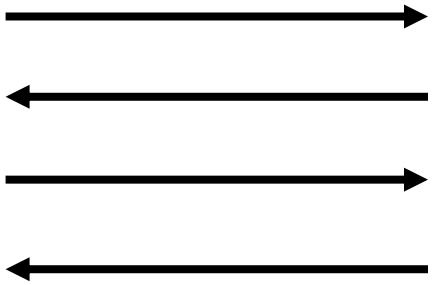
Charlie	Eric
Peihan	Frank
David	Melissa
Eve	Peihan
Melissa	Grace

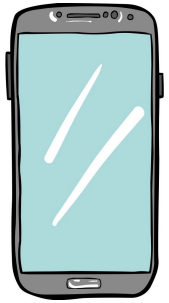




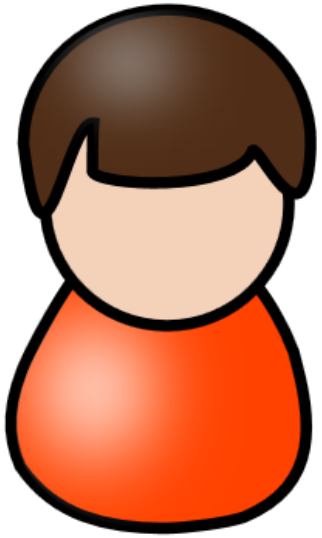


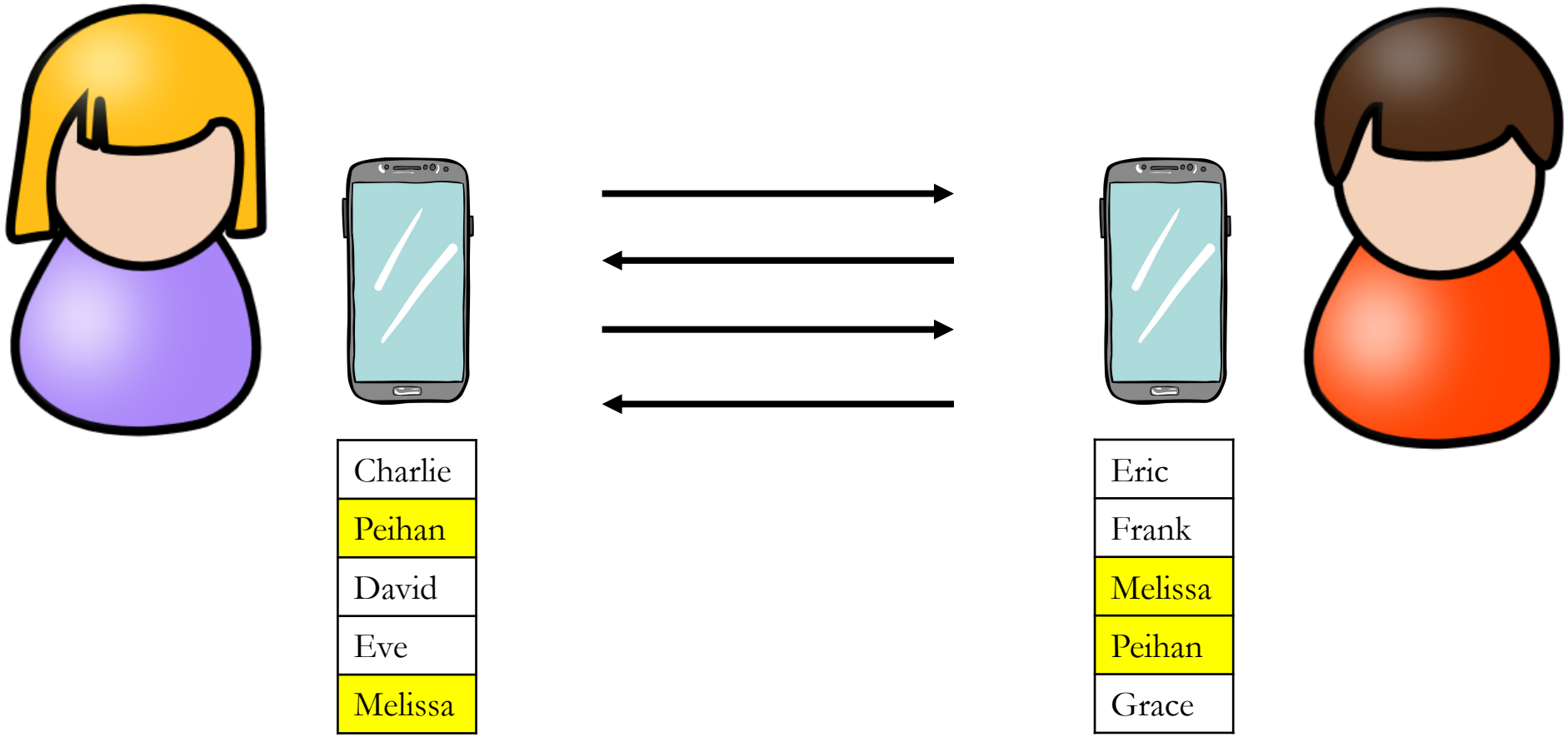
Charlie
Peihan
David
Eve
Melissa

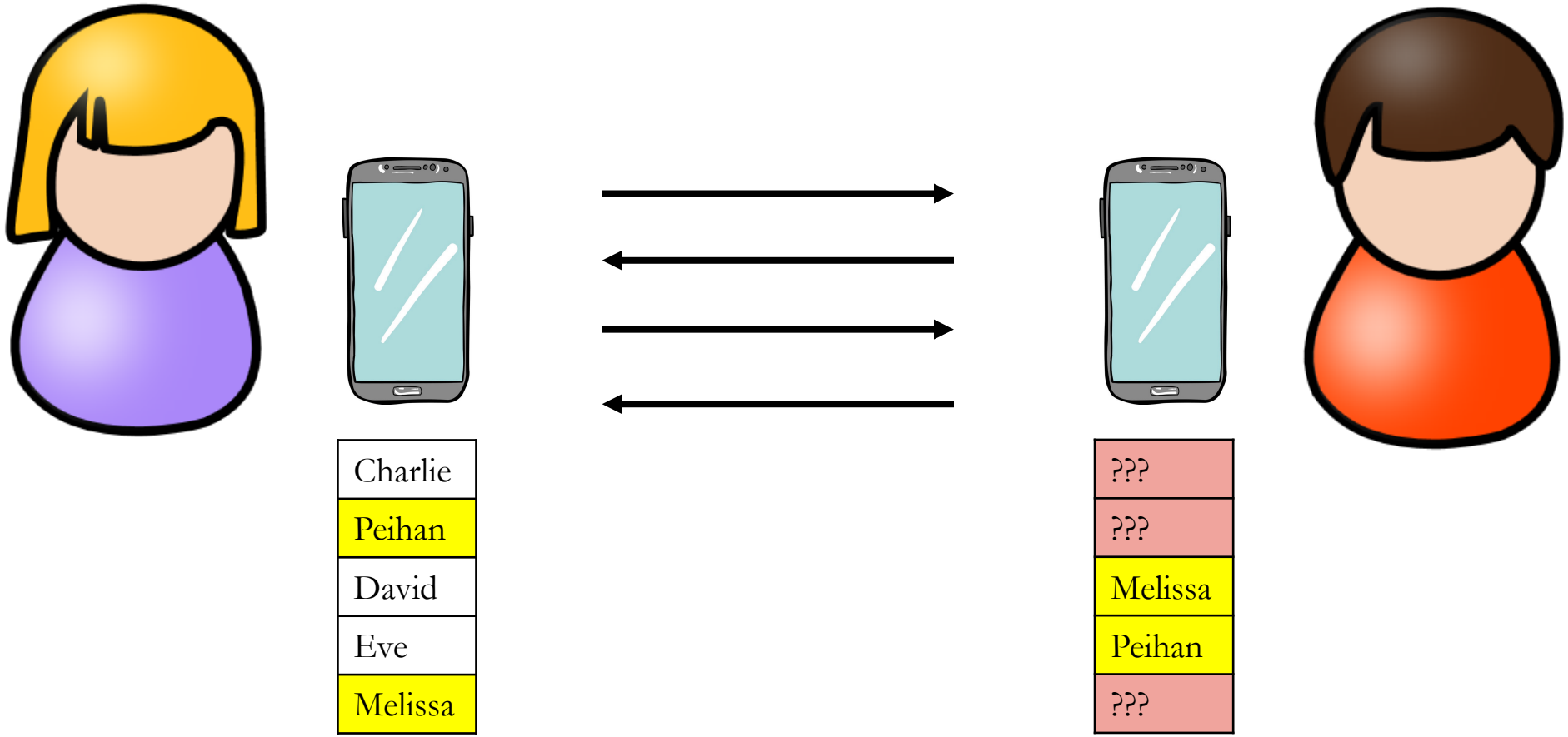


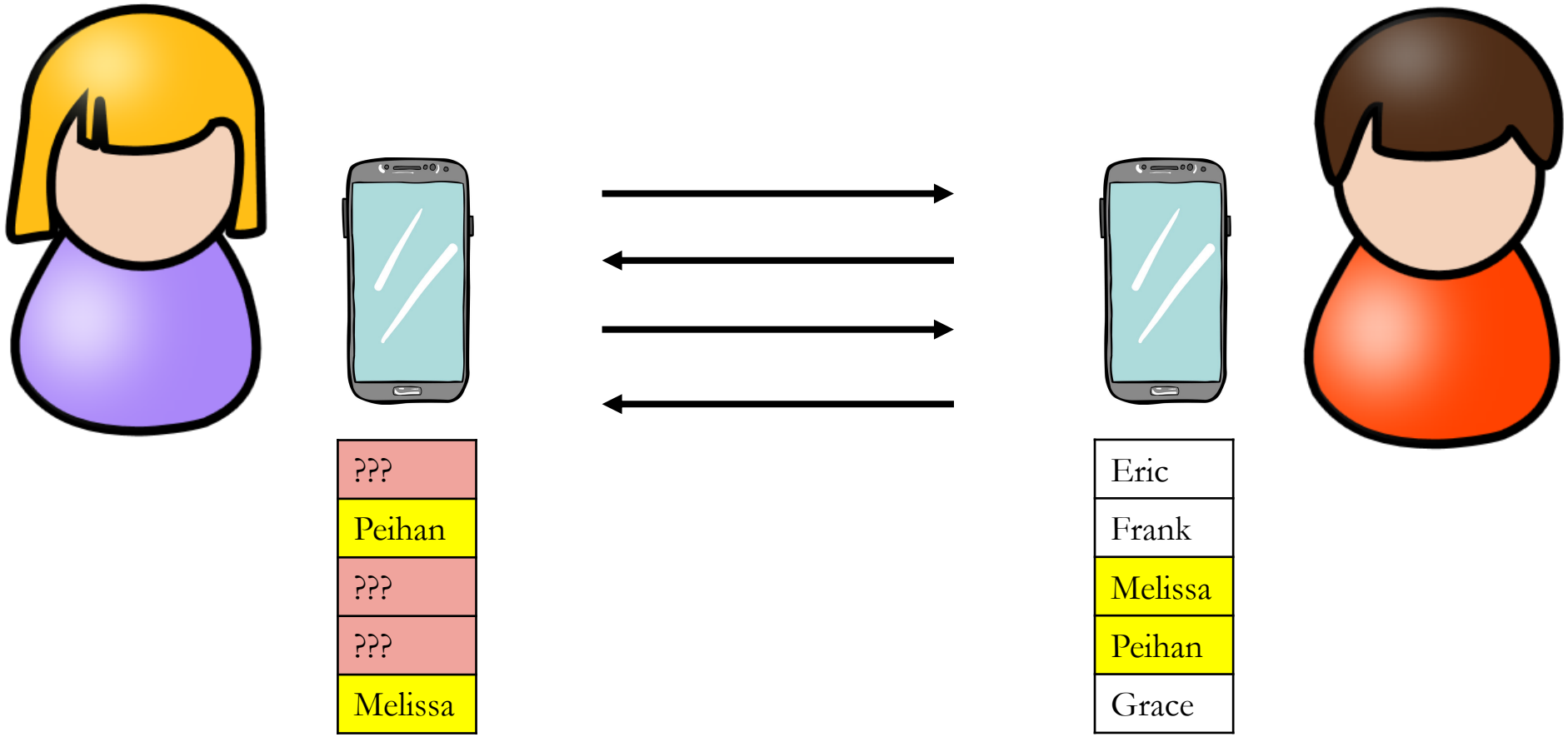


Eric
Frank
Melissa
Peihan
Grace

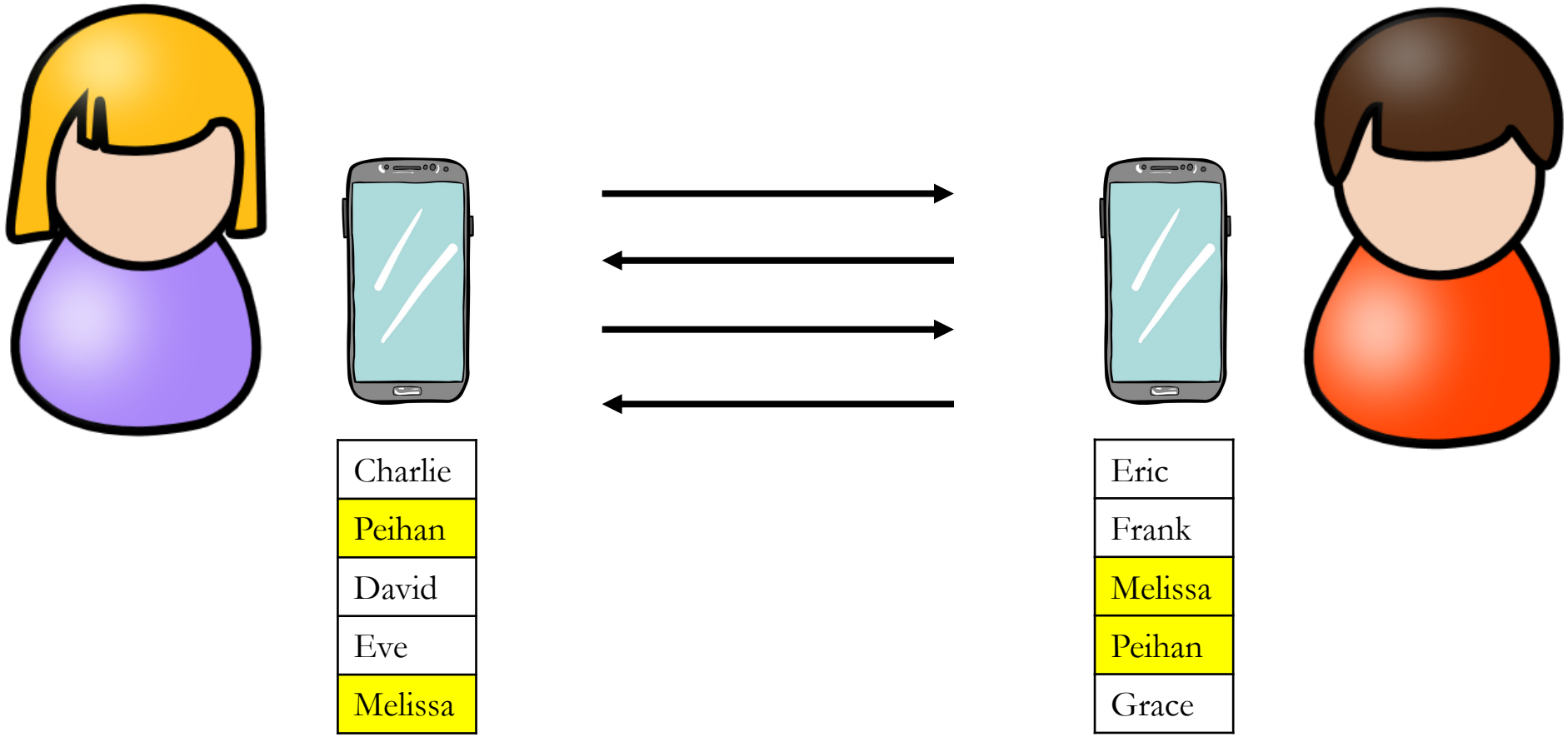



Charlie
Peihan
David
Eve
Melissa
Eric
Frank
Melissa
Peihan
Grace


Charlie
Peihan
David
Eve
Melissa
???
???
Melissa
Peihan
???


???
Peihan
???
???
Melissa
Eric
Frank
Melissa
Peihan
Grace

Private Set Intersection (PSI)



Google open-sources cryptographic tool to keep data sets private

by RAVIE LAKSHMANAN — Jun 20, 2019 in SECURITY

79 SHARES

https://tnw.to/C0PP1

Poorly secured databases are a top privacy and security concern — and Google now wants to plug that leak.

The internet giant has said it's open sourcing Private Join and Compute, a new secure multi-party computation (MPC) tool designed to help organizations work together with confidential data sets.

Ads Conversion Measurement

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Protect your accounts from data breaches with Password Checkup

February 5, 2019

Posted by Jennifer Pullman, Kurt Thomas, and Elie Bursztein, Security and Anti-abuse research

Update (Feb 6): We have updated the post to clarify a protocol used in the design is centered around private set intersection.

Google helps keep your account safe from hijacking with a defense in depth strategy that spans prevention, detection, and mitigation. As part of this, we regularly reset the passwords of Google accounts affected by third-party data breaches in the event of password reuse. This strategy has helped us protect over 110 million users in the last two years alone. Without these safety measures, users would be at ten times the risk of account hijacking.

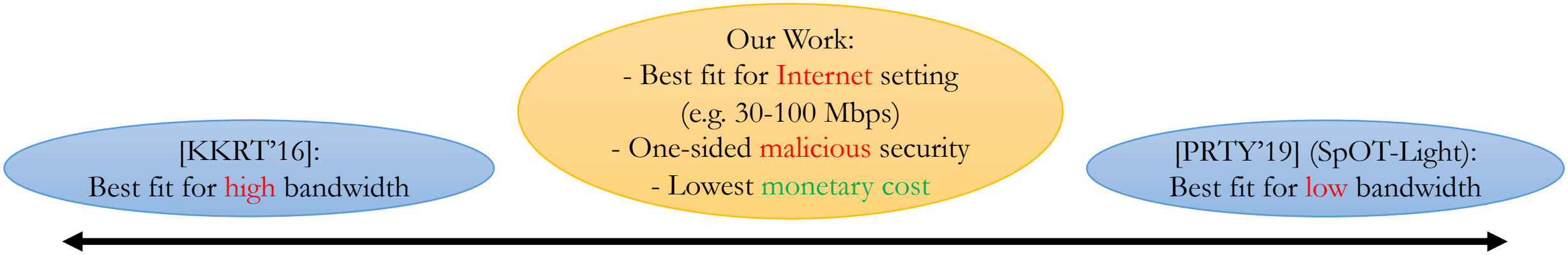
We want to help you stay safe not just on Google, but elsewhere on the web as well. This is where the new Password Checkup Chrome extension can help. Whenever you sign in to a site, Password Checkup will trigger a warning if the username and password you use is one of over 4 billion credentials that Google knows to be unsafe.

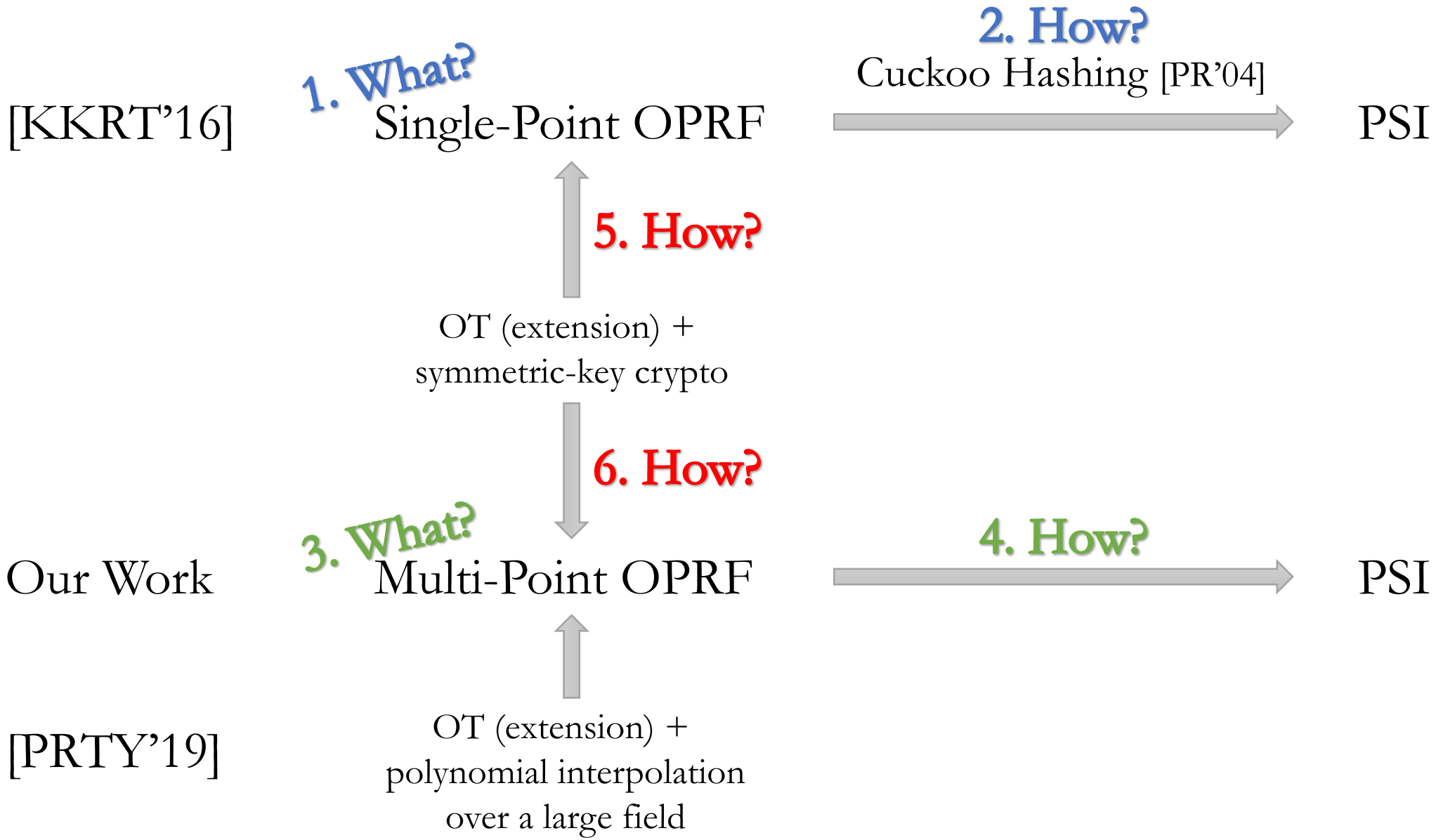
Password Breach Alert

State-of-the-art PSI

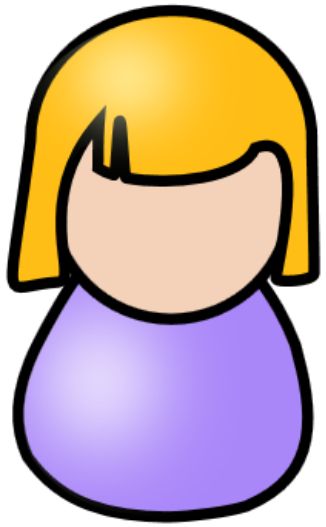
- Prior work [FNP'04, DSMRY'09, DCKT'10, ADCT'11, HEK'12, DCW13, PSZ14, PSSZ15, PSZ'14, KKRT'16, RR17a, RR17b, CLR17, DRRT18, PSWW18, GN'19, PRTY'19, PRTY'20, …]
- This work: semi-honest security, sets of the same size
- Which protocol should we adopt?
  - Most efficient one!
  - Tradeoff between Computation & Communication
  - Monetary cost [PRTY'19]

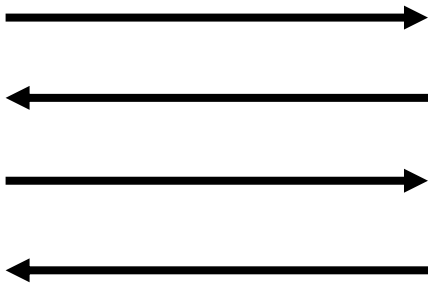
Can we achieve a better balance?

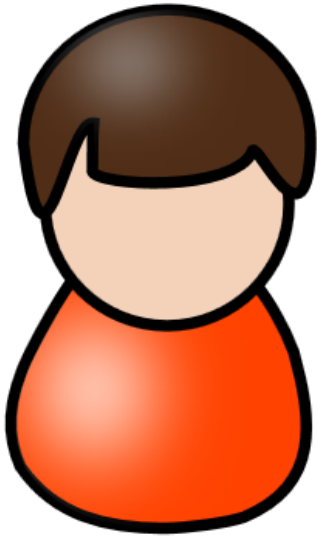



2. How?
Cuckoo Hashing [PR'04]
[KKRT'16]
1. What?
Single-Point OPRF
PSI
5. How?
OT (extension) +
symmetric-key crypto
6. How?
3. What?
4. How?
Our Work
Multi-Point OPRF
PSI
[PRTY'19]
OT (extension) +
polynomial interpolation
over a large field

Single-Point Oblivious Pseudorandom Function (OPRF)







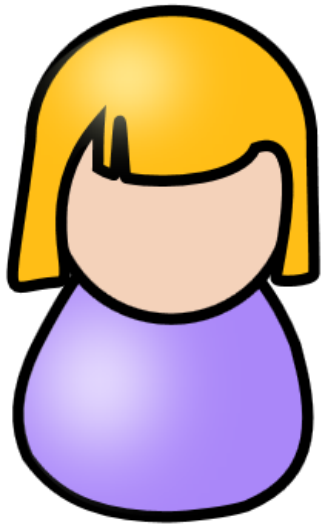
Input: $\perp$

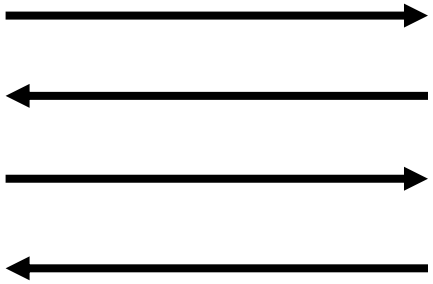
Output: k

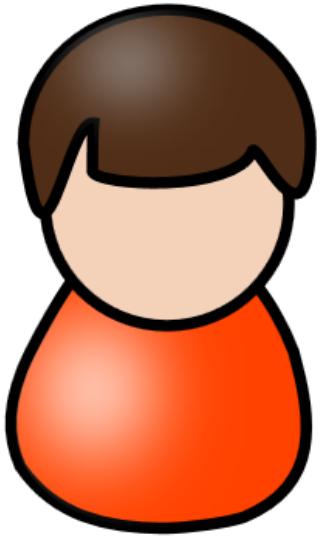
Input: y

Output: $F_k(y)$

Single-Point PSI from Single-Point OPRF







Input: $\perp$

Output: k

Input: y

Output: $F_k(y)$

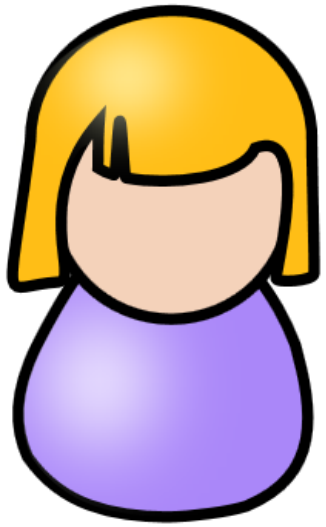
y

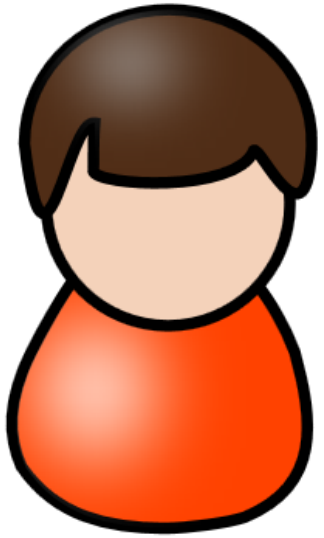
x_1
x_2
x_3
…
…

$F_k(x_1), F_k(x_2), F_k(x_3), \ldots$

$y \in X$?

PSI from Single-Point OPRF?



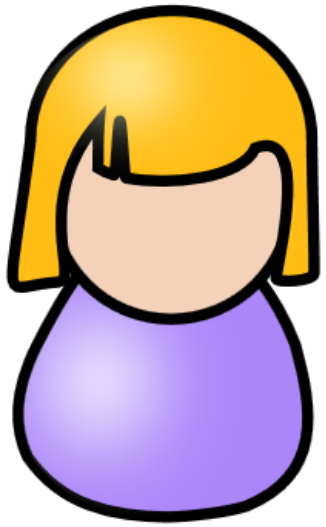


x_1
x_2
x_3
…
…

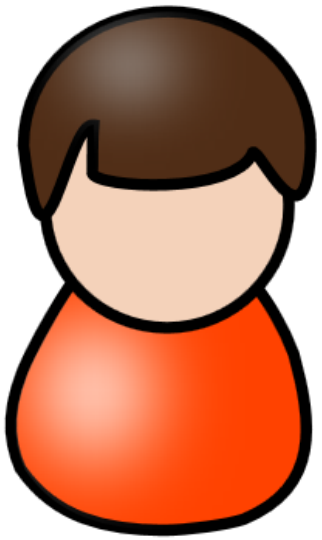
Computation & Communication: $O(n^2)$

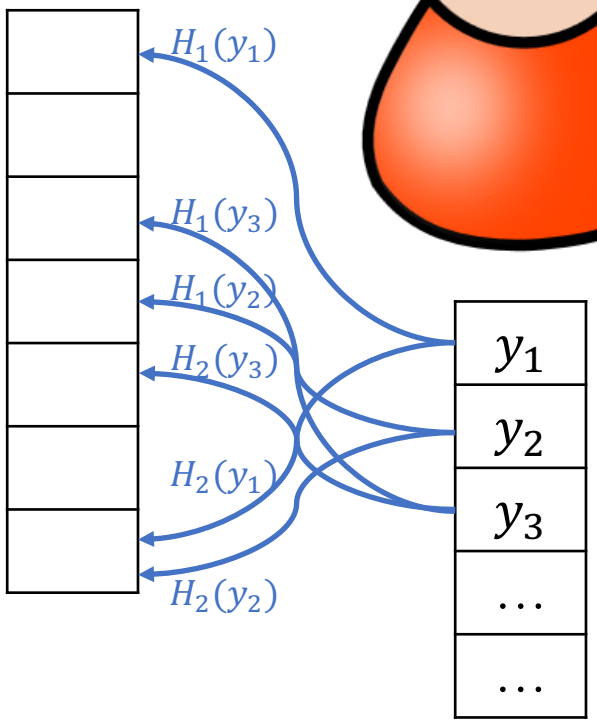
y_1	$y_1 \in X?$
y_2	$y_2 \in X?$
y_3	$y_3 \in X?$
…	…
…	…

PSI from Single-Point OPRF using Cuckoo Hashing [KKRT'16]

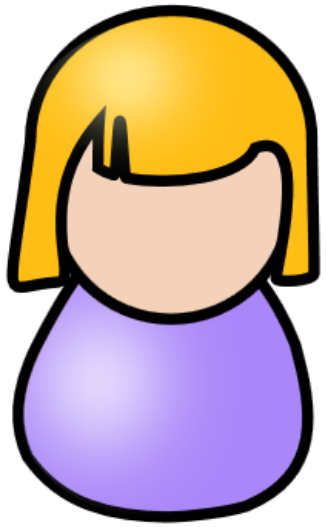


x_1
x_2
x_3
…
…

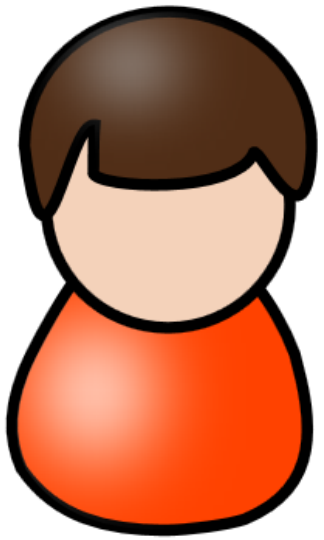




PSI from Single-Point OPRF using Cuckoo Hashing [KKRT'16]



x_1
x_2
x_3
…
…

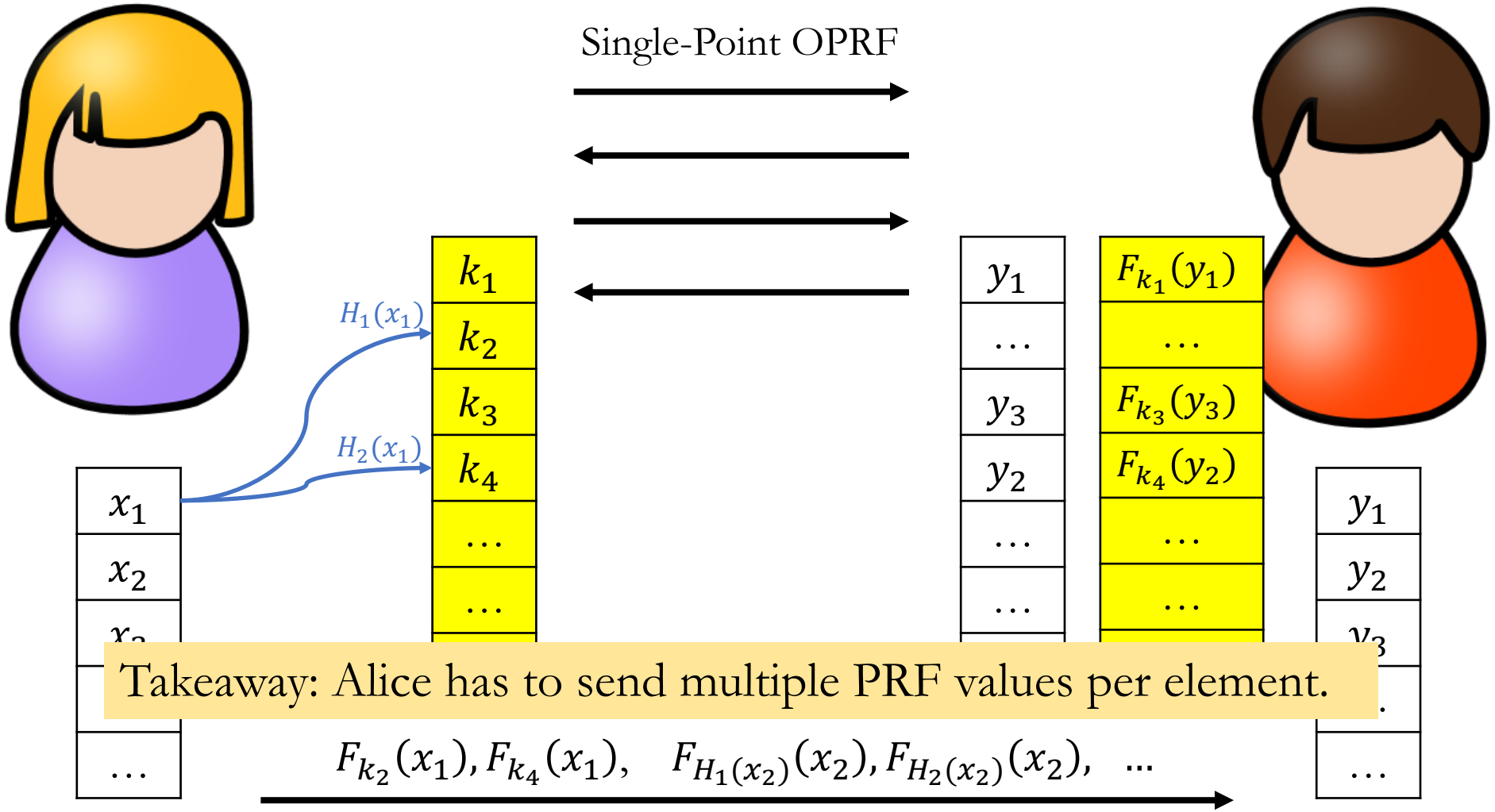


y_1
…
y_3
y_2
…
…
…

$H_1(y_1)$
$H_1(y_3)$
$H_1(y_2)$
$H_2(y_3)$
$H_2(y_1)$
$H_2(y_2)$

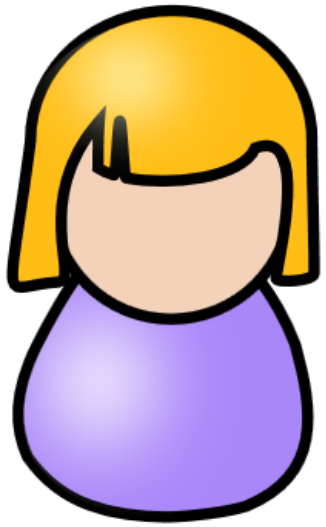
y_1
y_2
y_3
…
…

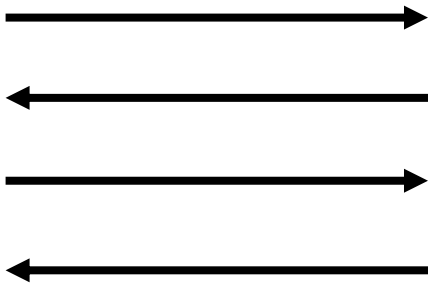
PSI from Single-Point OPRF using Cuckoo Hashing [KKRT'16]

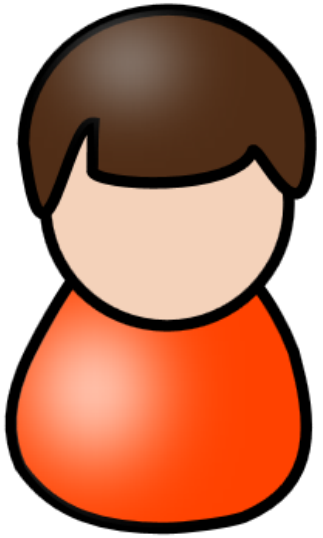


Takeaway: Alice has to send multiple PRF values per element.

Single-Point OPRF







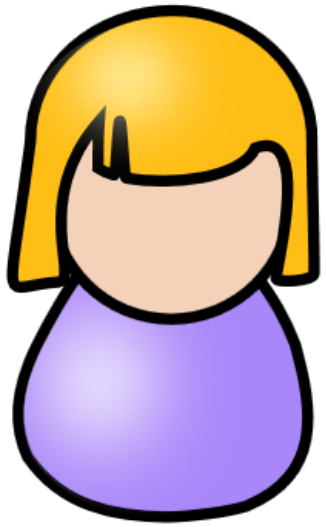
Input: $\perp$

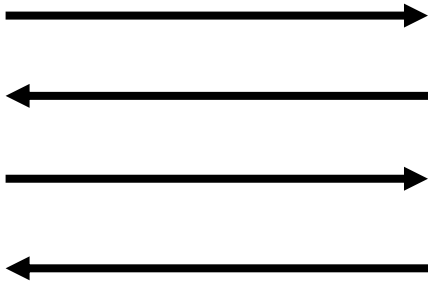
Output: k

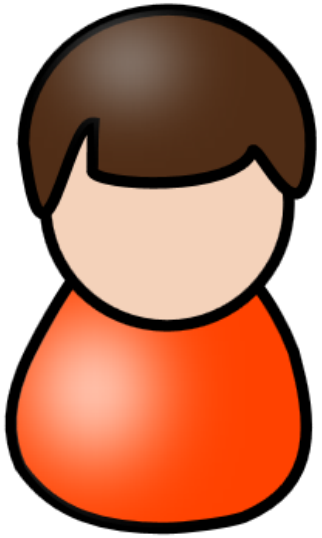
Input: y

Output: $F_k(y)$

Multi-Point OPRF







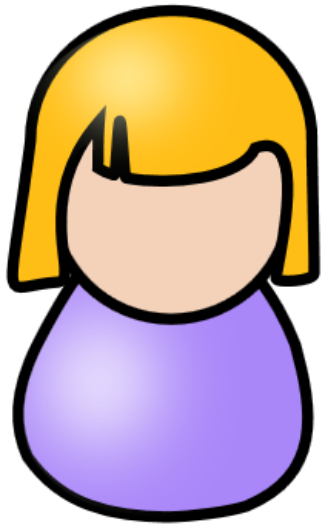
Input: $\perp$

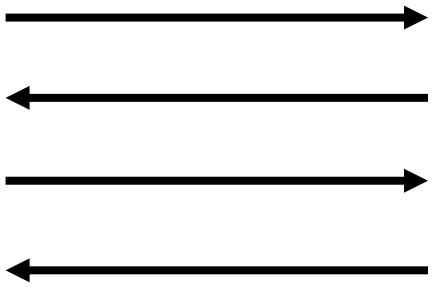
Output: k

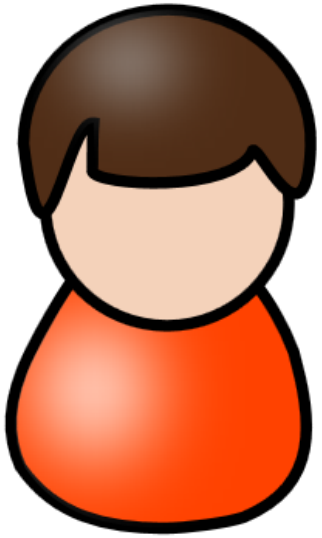
Input: $y_1, y_2, \ldots, y_n$

Output: $F_k(y_1), \ldots, F_k(y_n)$

PSI from Multi-Point OPRF







Input: $\perp$

Output: k

Input: $y_1, y_2, \ldots, y_n$

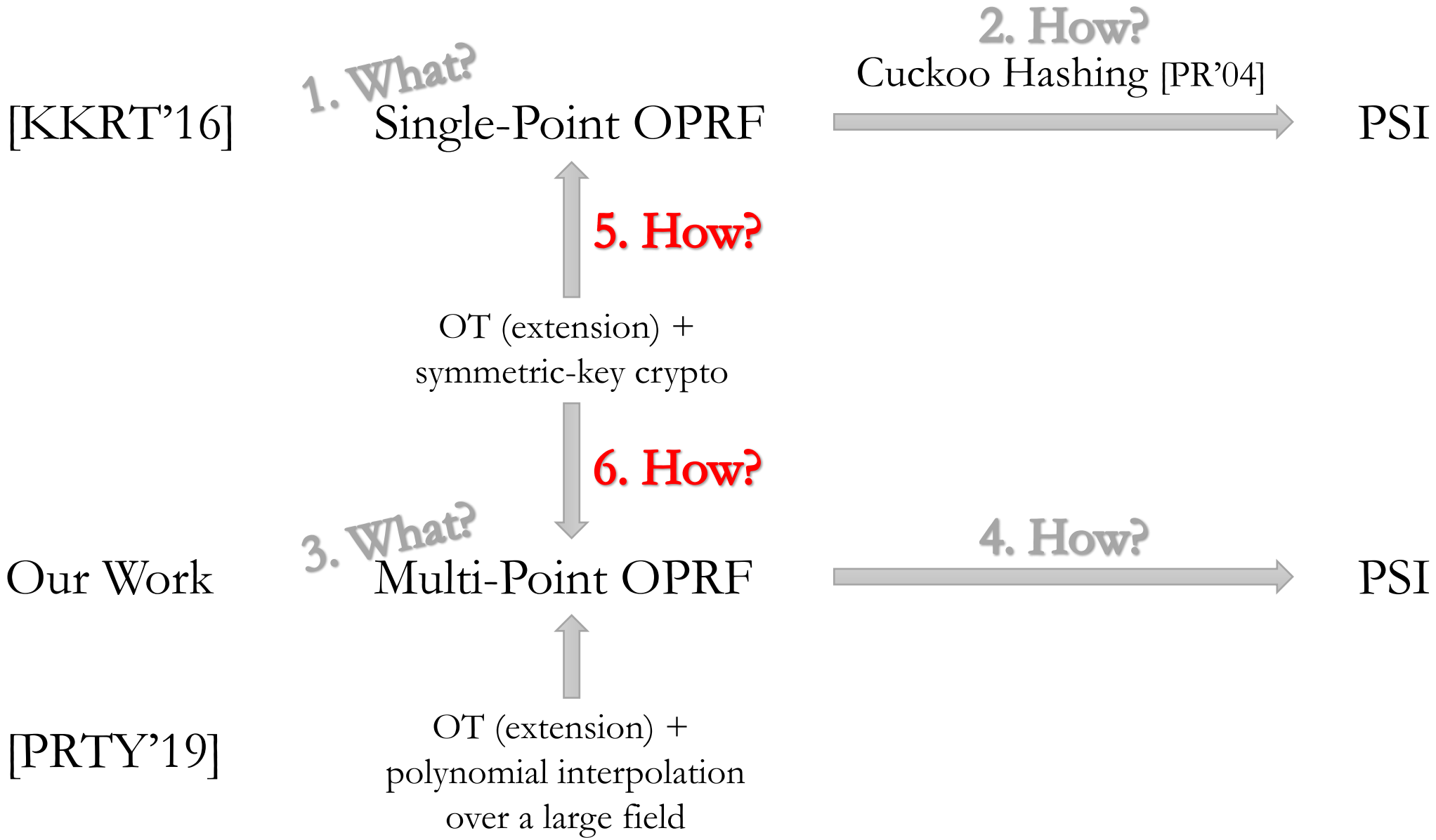
Output: $F_k(y_1), \ldots, F_k(y_n)$

x_1
x_2
x_3
…
…

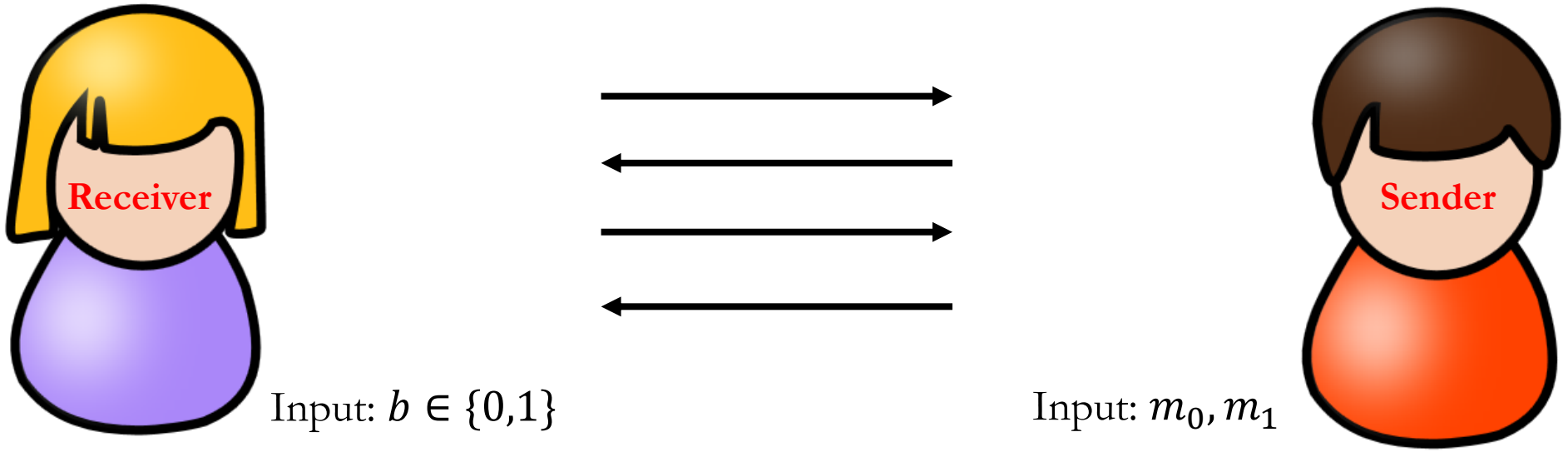
$F_k(x_1), F_k(x_2), F_k(x_3), \ldots$

y_1
y_2
y_3
…
…

$X \cap Y$

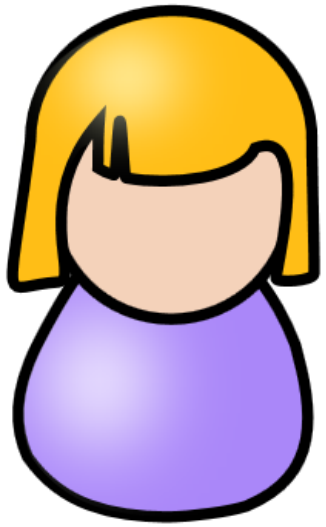

1. What?
2. How?
Cuckoo Hashing [PR'04]
[KKRT'16]
Single-Point OPRF
PSI
5. How?
OT (extension) +
symmetric-key crypto
6. How?
3. What?
4. How?
Our Work
Multi-Point OPRF
PSI
[PRTY'19]
OT (extension) +
polynomial interpolation
over a large field

Oblivious Transfer (OT) [Rabin'05]



Note: N OTs can be done by $O(\lambda)$ public-key operations and $O(N)$ symmetric-key operations using OT extension [IKNP'03].

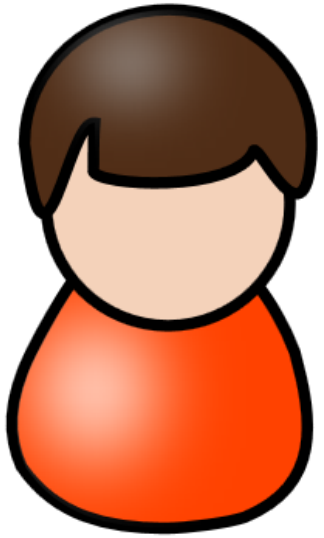
Single-Point OPRF [KKRT'16]



Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$



Input: y

0	1	0	0	1

$r_0 \leftarrow \{0,1\}^{\lambda}$

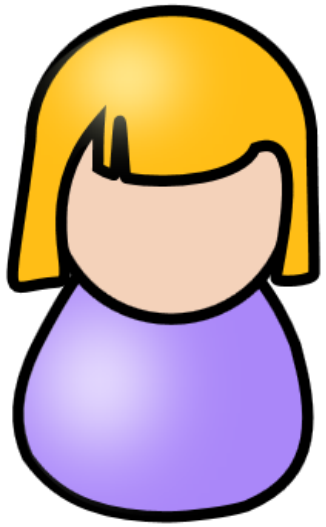
$r_1 \in \{0,1\}^{\lambda}$

$y \xrightarrow{f(y)}$

1	1	0	1	0

pseudorandom, deterministic,
e.g. $PRF_k(y)$ with public k

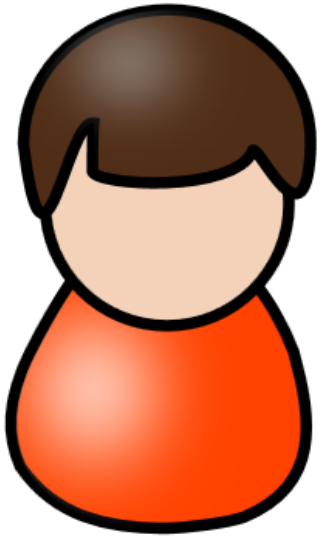
Single-Point OPRF [KKRT'16]



Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$



Input: y

0	1	0	0	1

$r_0 \leftarrow \{0,1\}^{\lambda}$

1	0	0	1	1

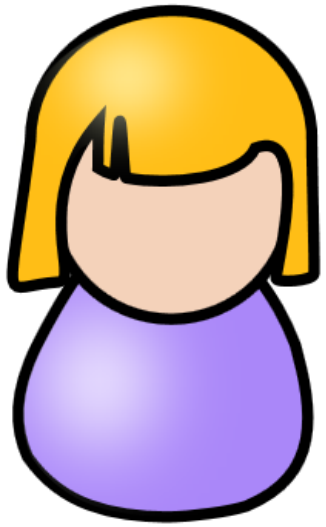
$r_1 = r_0 \oplus f(y)$

$y \xrightarrow{f(y)}$

1	1	0	1	0

pseudorandom, deterministic,
e.g. $PRF_k(y)$ with public k

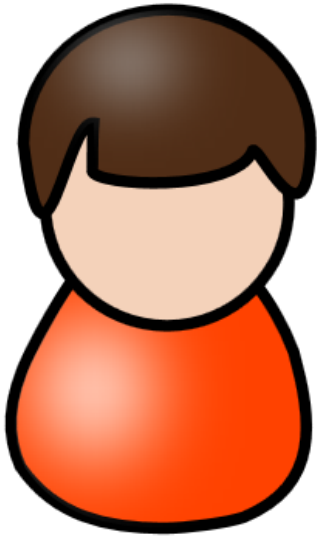
Single-Point OPRF [KKRT'16]



Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$



Input: y

0	1	0	0	1

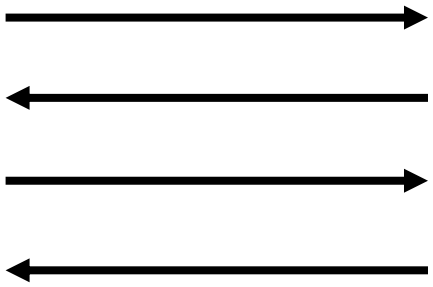
$r_0 \leftarrow \{0,1\}^{\lambda}$

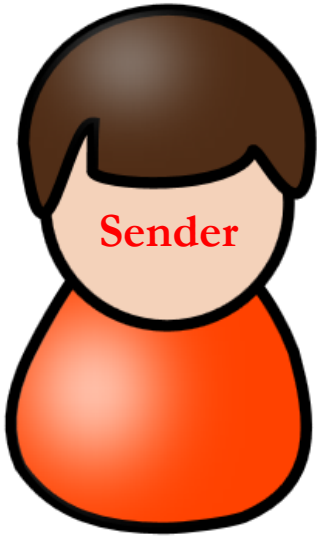
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

Single-Point OPRF [KKRT'16]

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: y

0	1	0	0	1

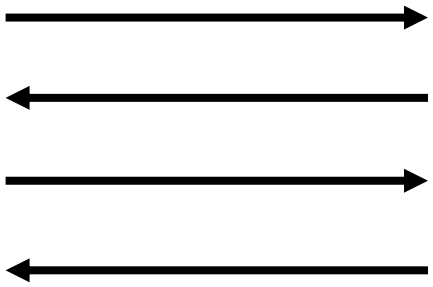
$r_0 \leftarrow \{0,1\}^{\lambda}$

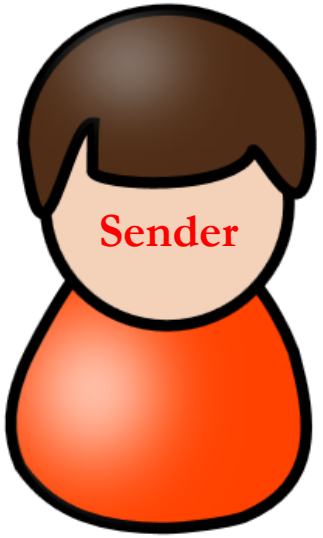
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

Single-Point OPRF [KKRT'16]

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: y

0	1	0	0	1

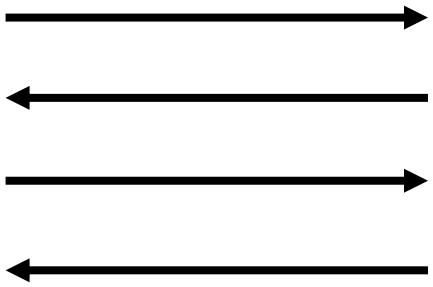
$r_0 \leftarrow \{0,1\}^{\lambda}$

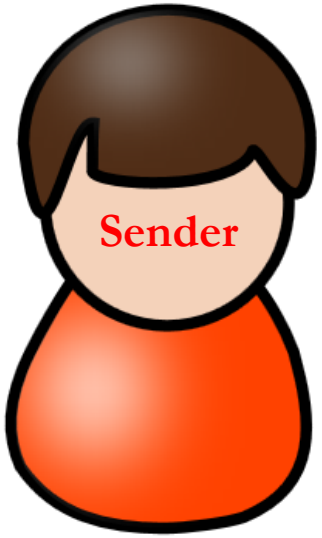
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

Single-Point OPRF [KKRT'16]

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: y

0	1	0	0	1

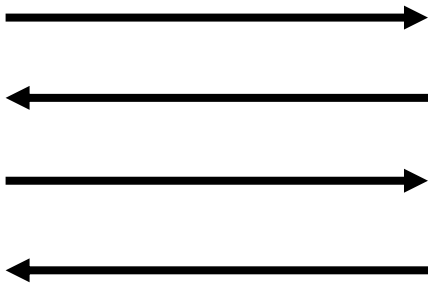
$r_0 \leftarrow \{0,1\}^{\lambda}$

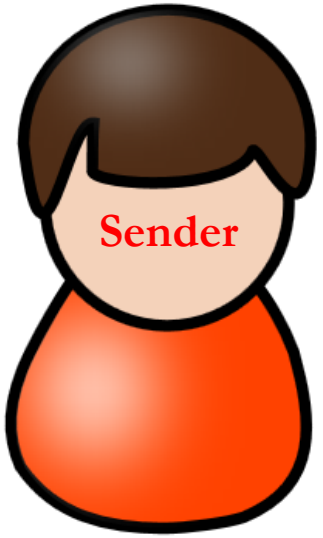
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

Single-Point OPRF [KKRT'16]

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: y

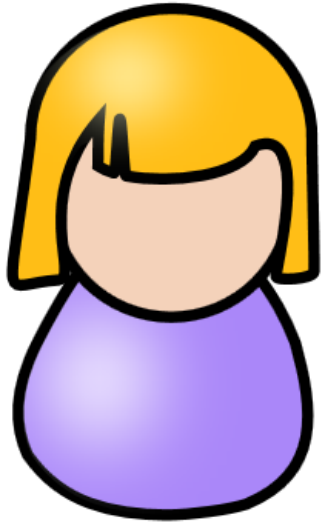
0	1	0	0	1

$r_0 \leftarrow \{0,1\}^{\lambda}$

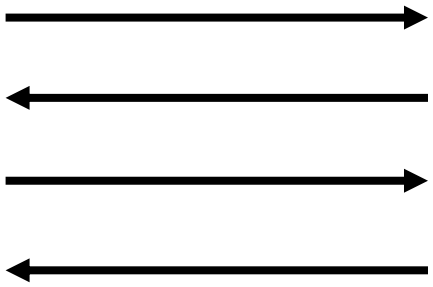
1	0	0	1	1

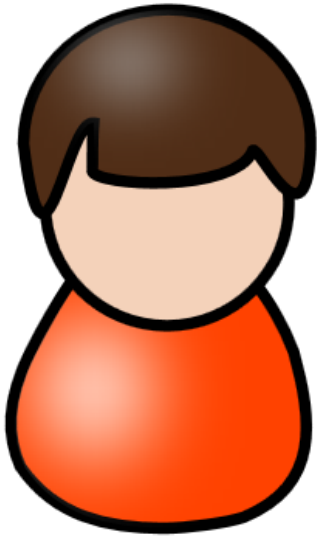
$r_1 = r_0 \oplus f(y)$

Single-Point OPRF [KKRT'16]



OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

1	1	0	1	1

$q = r_0 \oplus [s \cdot f(y)]$

$x \xrightarrow{f(x)}$

0	1	0	0	1

Input: y

0	1	0	0	1

$r_0 \leftarrow \{0,1\}^{\lambda}$

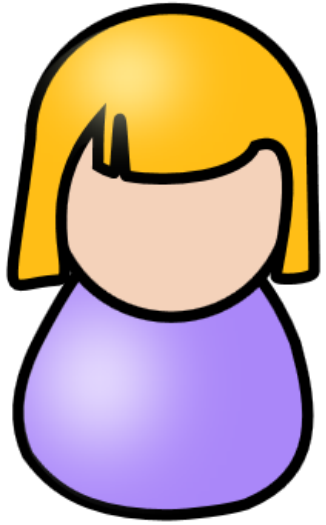
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

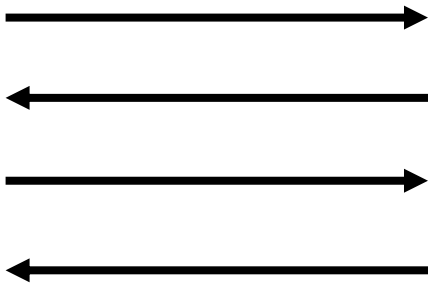
Output: $k = (s, q),\ \ F_k(x) = H(q \oplus [s \cdot f(x)])$

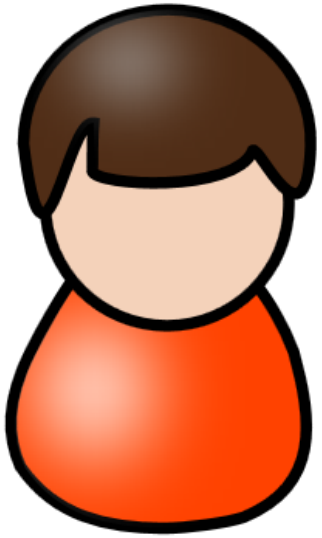
Output: $F_k(y) = H(r_0)$

Single-Point OPRF [KKRT'16]



OT





Input: $\perp$

Input: y

1	0	0	1	0

$s \leftarrow \{0,1\}^\lambda$

1	1	0	1	1

$q = r_0 \oplus [s \cdot f(y)]$

$x \xrightarrow{f(x)}$

0	1	0	0	1

0	1	0	0	1

$r_0 \leftarrow \{0,1\}^\lambda$

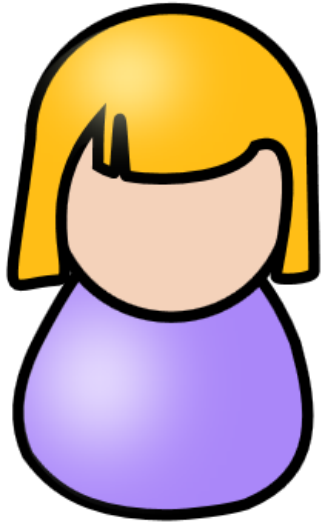
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

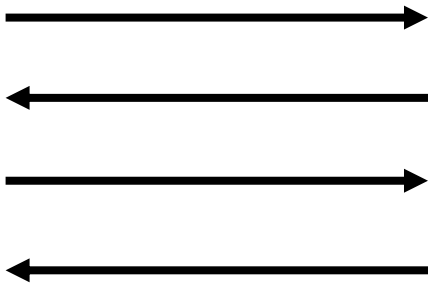
Output: $k = (s, q),\ \ F_k(x) = H(q \oplus [s \cdot f(x)])$

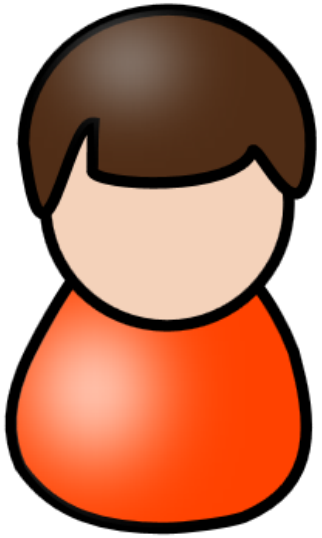
Output: $F_k(y) = H(r_0)$

Single-Point OPRF [KKRT'16]



OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^\lambda$

1	1	0	1	1

$q = r_0 \oplus [s \cdot f(y)]$

$x \xrightarrow{f(x)}$

0	1	0	0	1

Input: y

0	1	0	0	1

$r_0 \leftarrow \{0,1\}^\lambda$

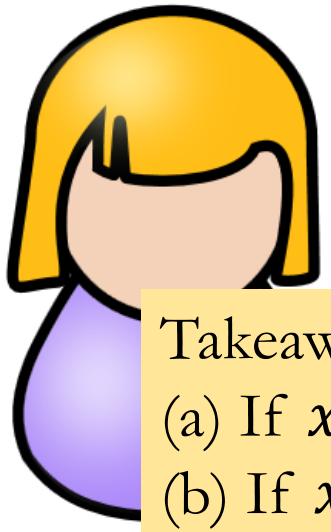
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

Output: $k = (s, q),\ \ F_k(x) = H(q \oplus [s \cdot f(x)])$

Output: $F_k(y) = H(r_0)$

Single-Point OPRF [KKRT'16]



OT

Takeaways:
(a) If $x = y$, then $F_k(x) = H(r_0)$ no matter what s is chosen.
(b) If $x \neq y$, then $F_k(x)$ is hard to guess.

1	0	0	1	0

$s \leftarrow \{0,1\}^\lambda$

1	1	0	1	1

$q = r_0 \oplus [s \cdot f(y)]$

$x \xrightarrow{f(x)}$

0	1	0	0	1

0	1	0	0	1

$r_0 \leftarrow \{0,1\}^\lambda$

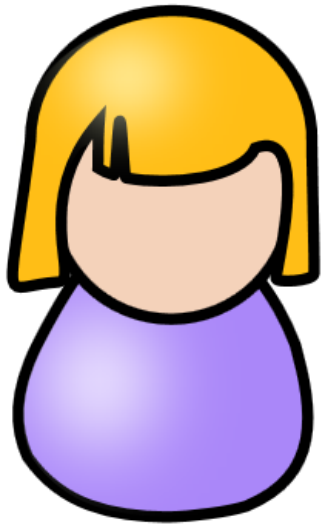
1	0	0	1	1

$r_1 = r_0 \oplus f(y)$

Output: $k = (s, q)$, $F_k(x) = H(q \oplus [s \cdot f(x)])$
$= H(r_0 \oplus [s \cdot (f(x) \oplus f(y)])$

Output: $F_k(y) = H(r_0)$

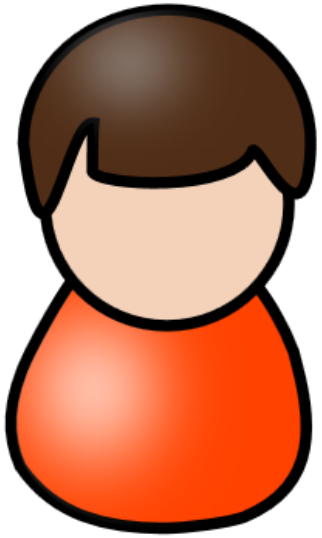
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

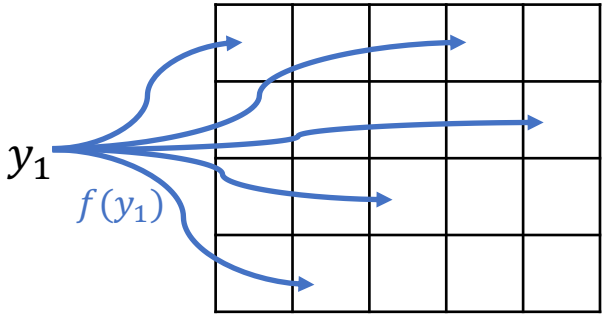
$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

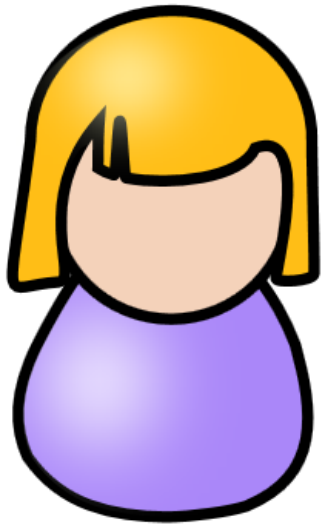
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$



$R_1 \in \{0,1\}^{n \times \lambda}$

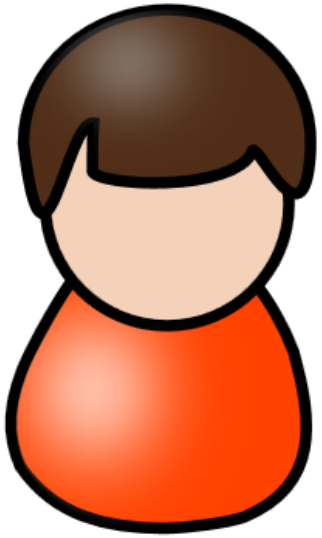
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

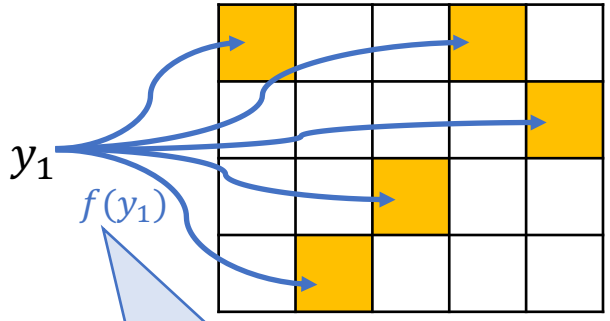
$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

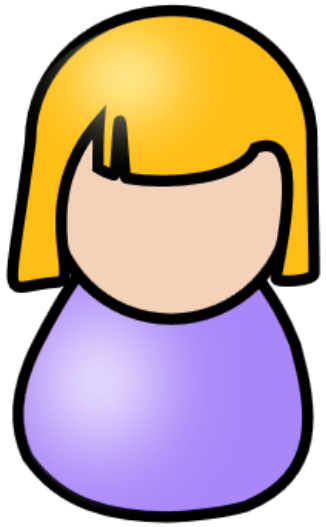
$R_0 \leftarrow \{0,1\}^{n \times \lambda}$



$R_1 \in \{0,1\}^{n \times \lambda}$

pseudorandom, deterministic,
e.g. $PRF_k(y)$ with public k

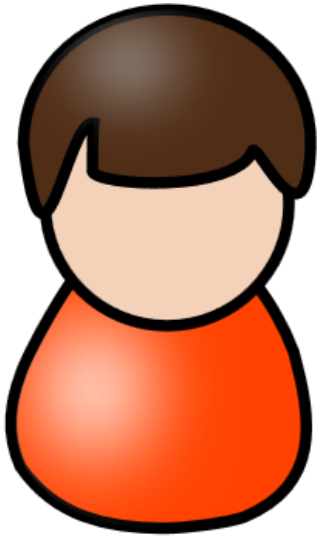
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

y_1

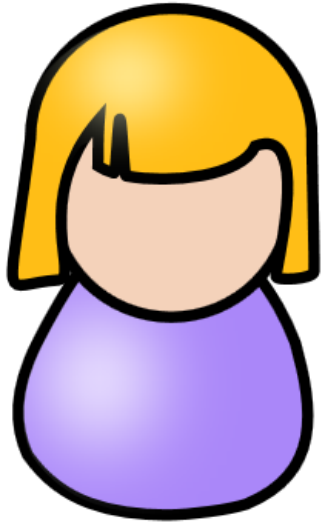
$f(y_1)$

0			1	
				0
		0		
	0			

$R_1 \in \{0,1\}^{n \times \lambda}$

pseudorandom, deterministic,
e.g. $PRF_k(y)$ with public k

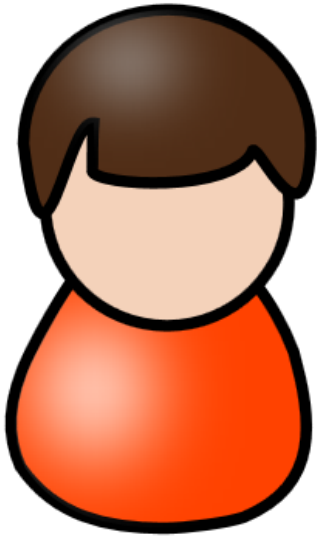
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

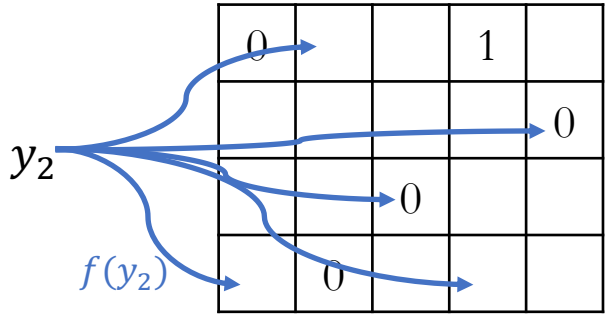
$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

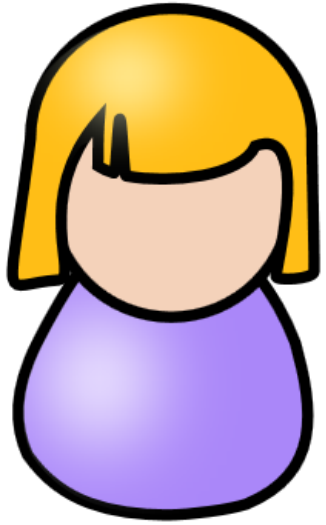
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$



$R_1 \in \{0,1\}^{n \times \lambda}$

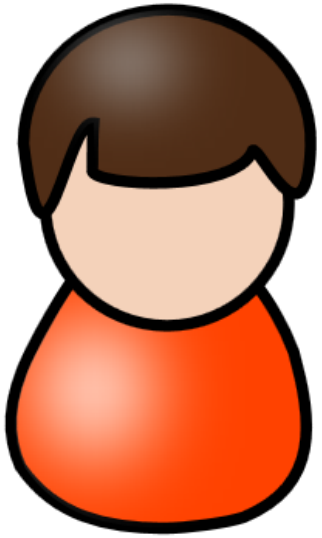
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

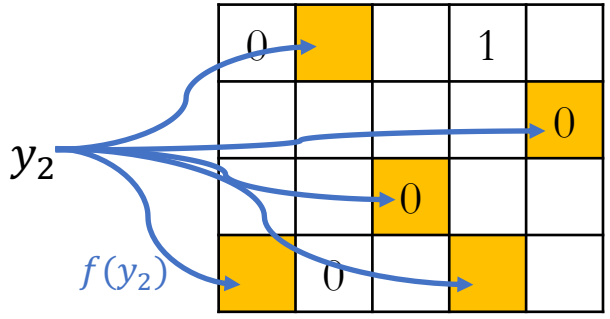
$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

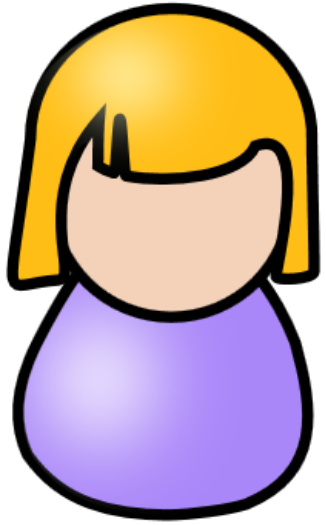
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$



$R_1 \in \{0,1\}^{n \times \lambda}$

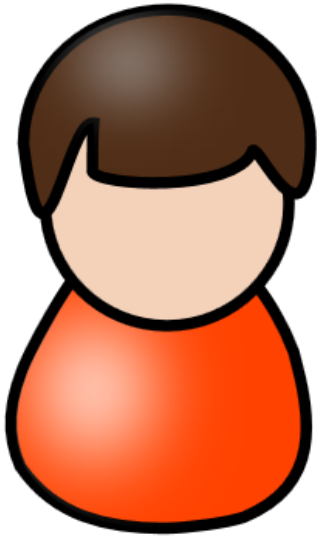
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

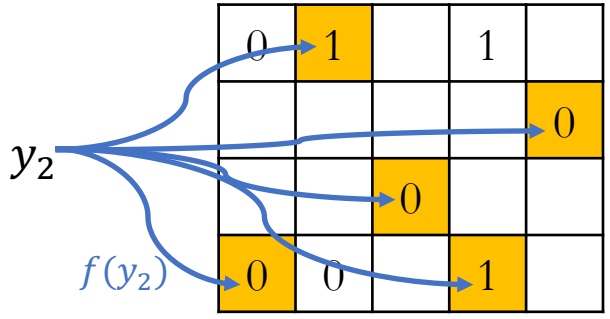
$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

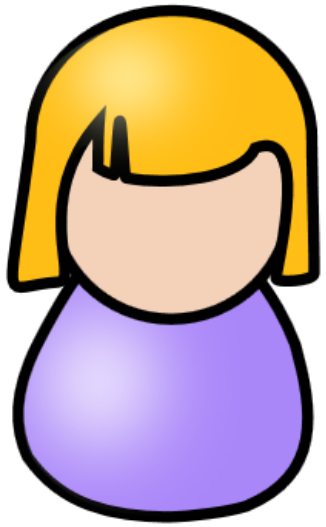
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$



$R_1 \in \{0,1\}^{n \times \lambda}$

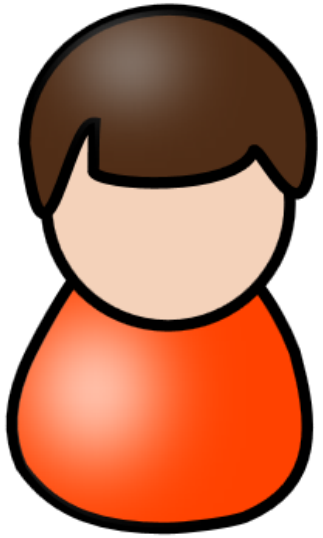
Multi-Point OPRF



Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$



Input: $y_1, y_2, \ldots, y_n$

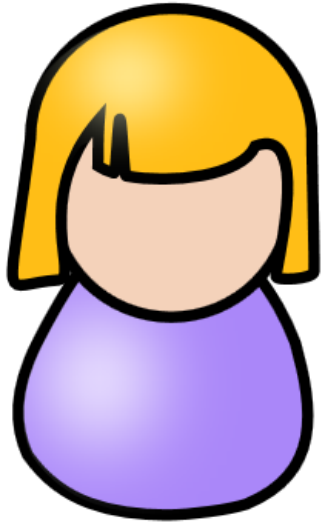
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

0	1		1	
				0
		0		
0	0		1	

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

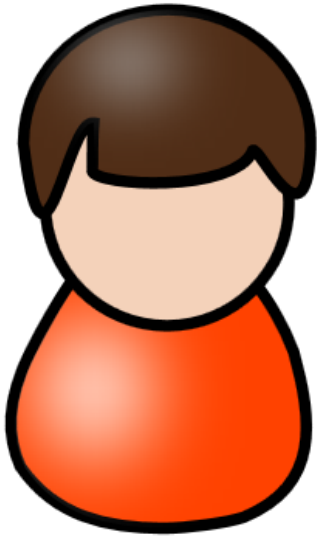


Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$



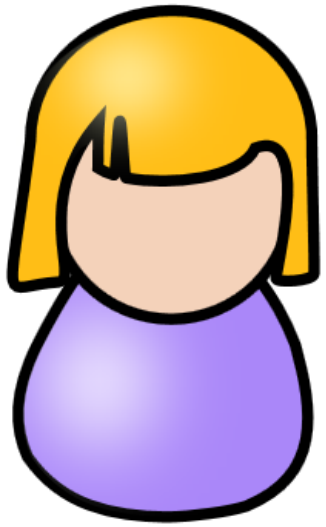
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

0	1	0	1	
1			1	0
		0		1
0	0		1	

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

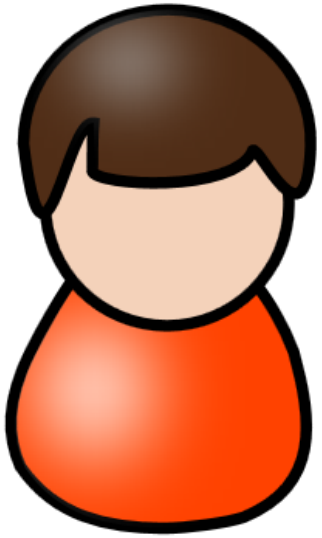


Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$



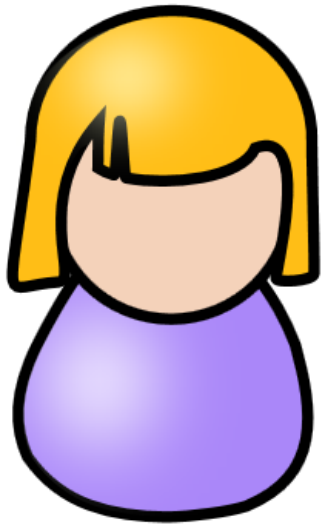
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

0	1	0	1	
1			1	0
		0		1
0	0		1	

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

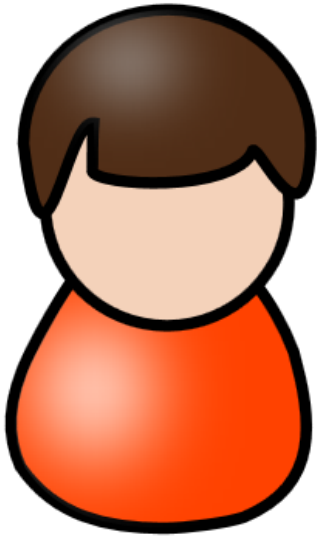


Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$



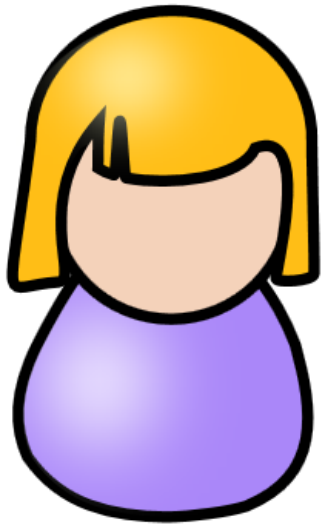
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

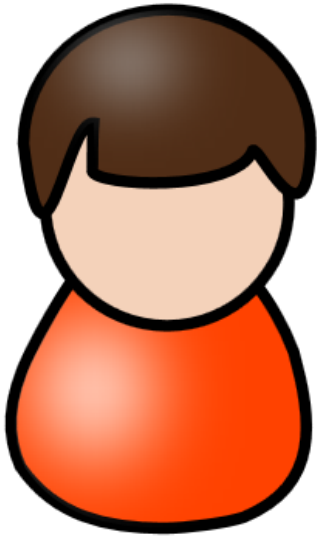


Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$



0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

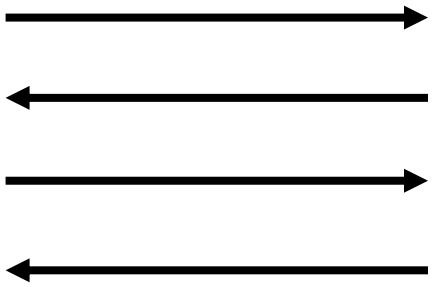
$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

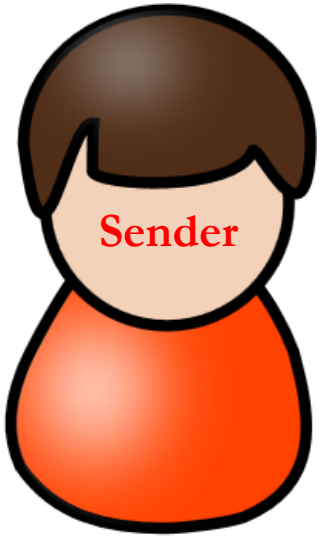
0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

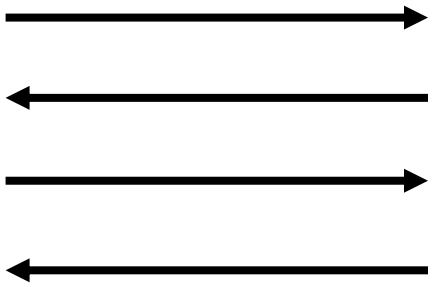
$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

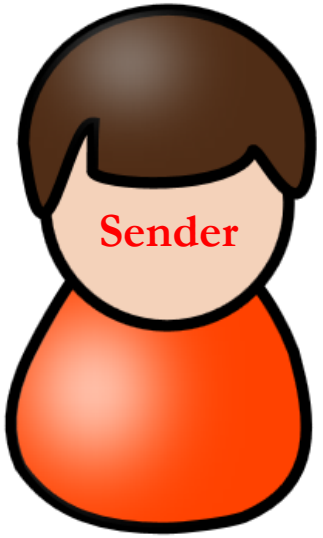
0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

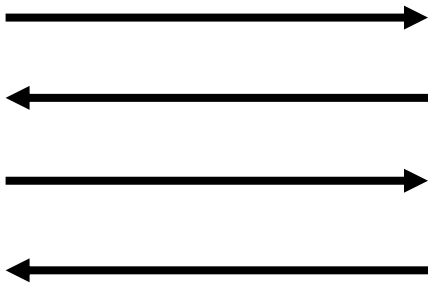
$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

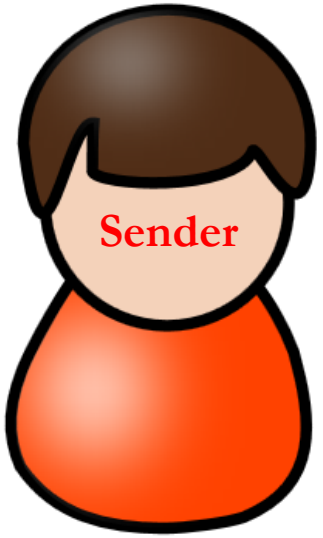
0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

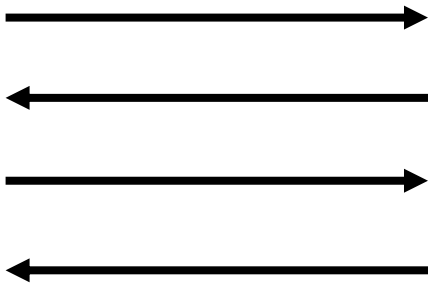
$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

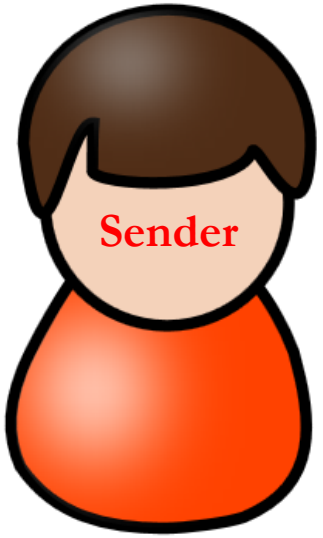
0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF

OT





Input: $\perp$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

Input: $y_1, y_2, \ldots, y_n$

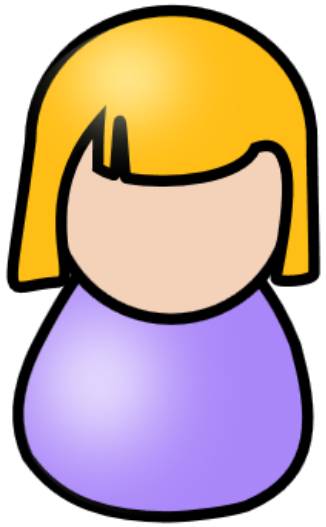
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

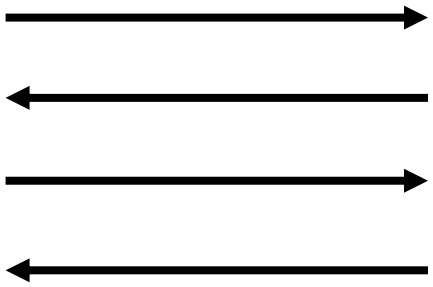
0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

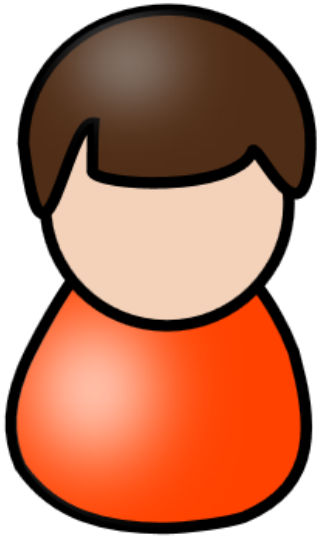
$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF



OT





Input: $\perp$

Input: $y_1, y_2, \ldots, y_n$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

$F_k(x)$:

x

$f(x)$

0	0	0	1	1
1	1	0	1	0
0	1	0	1	1
0	0	1	1	1

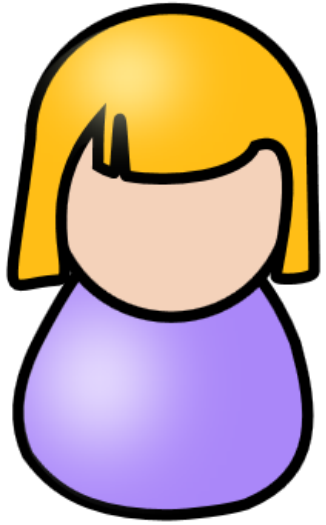
0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

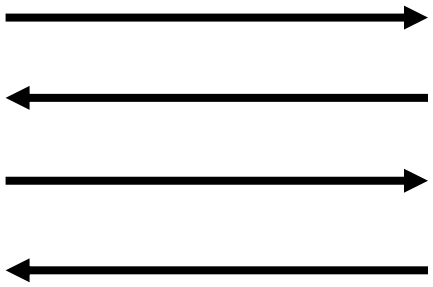
0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

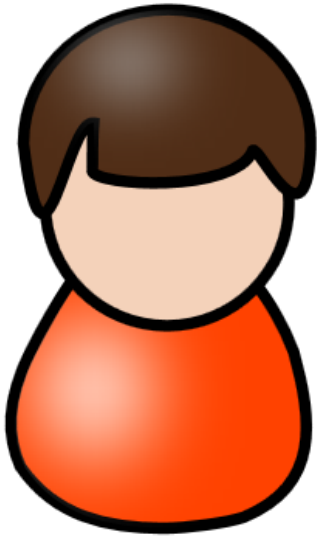
$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF



OT





Input: $\perp$

Input: $y_1, y_2, \dots, y_n$

1	0	0	1	0

$s \leftarrow \{0,1\}^{\lambda}$

$F_k(x)$:

x

$f(x)$

0	0	0	1	1
1	1	0	1	0
0	1	0	1	1
0	0	1	1	1

$H($ $)$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

$F_k(y_i)$:

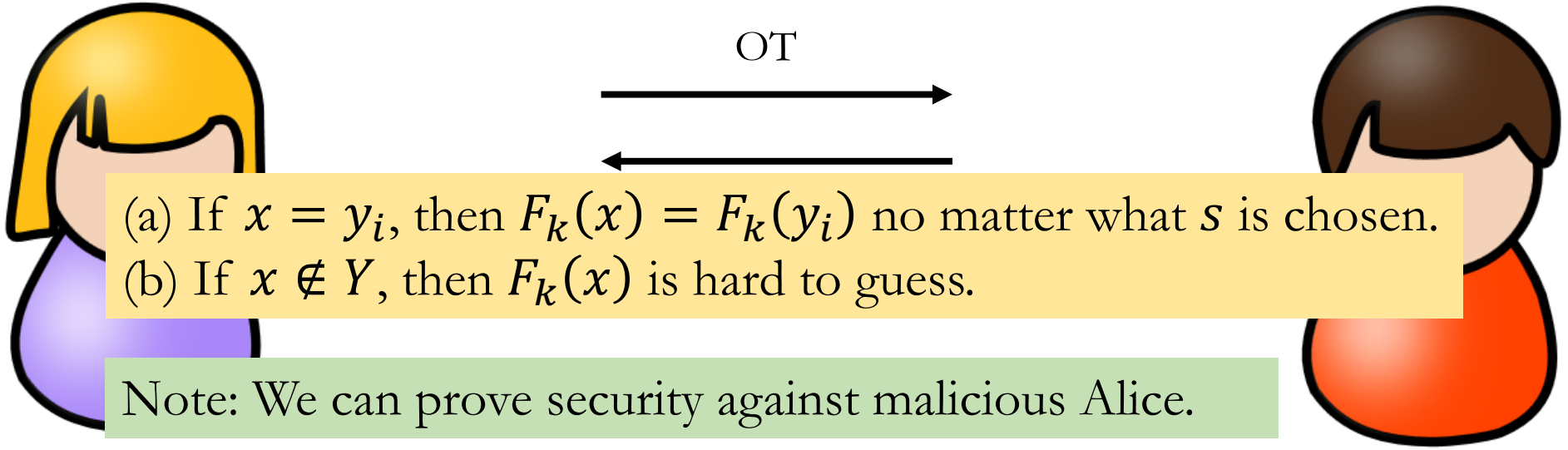
y_i

$f(y_i)$

0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

$R_1 \in \{0,1\}^{n \times \lambda}$

Multi-Point OPRF



(a) If $x = y_i$, then $F_k(x) = F_k(y_i)$ no matter what s is chosen.
(b) If $x \notin Y$, then $F_k(x)$ is hard to guess.

Note: We can prove security against malicious Alice.

$s \leftarrow \{0,1\}^\lambda$

1	0	0	1	0

$F_k(x)$:

x → $f(x)$

0	0	0	1	1
1	1	0	1	0
0	1	0	1	1
0	0	1	1	1

$H($ 0 0 0 1 0 $)$

0	1	0	1	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1

$R_0 \leftarrow \{0,1\}^{n \times \lambda}$

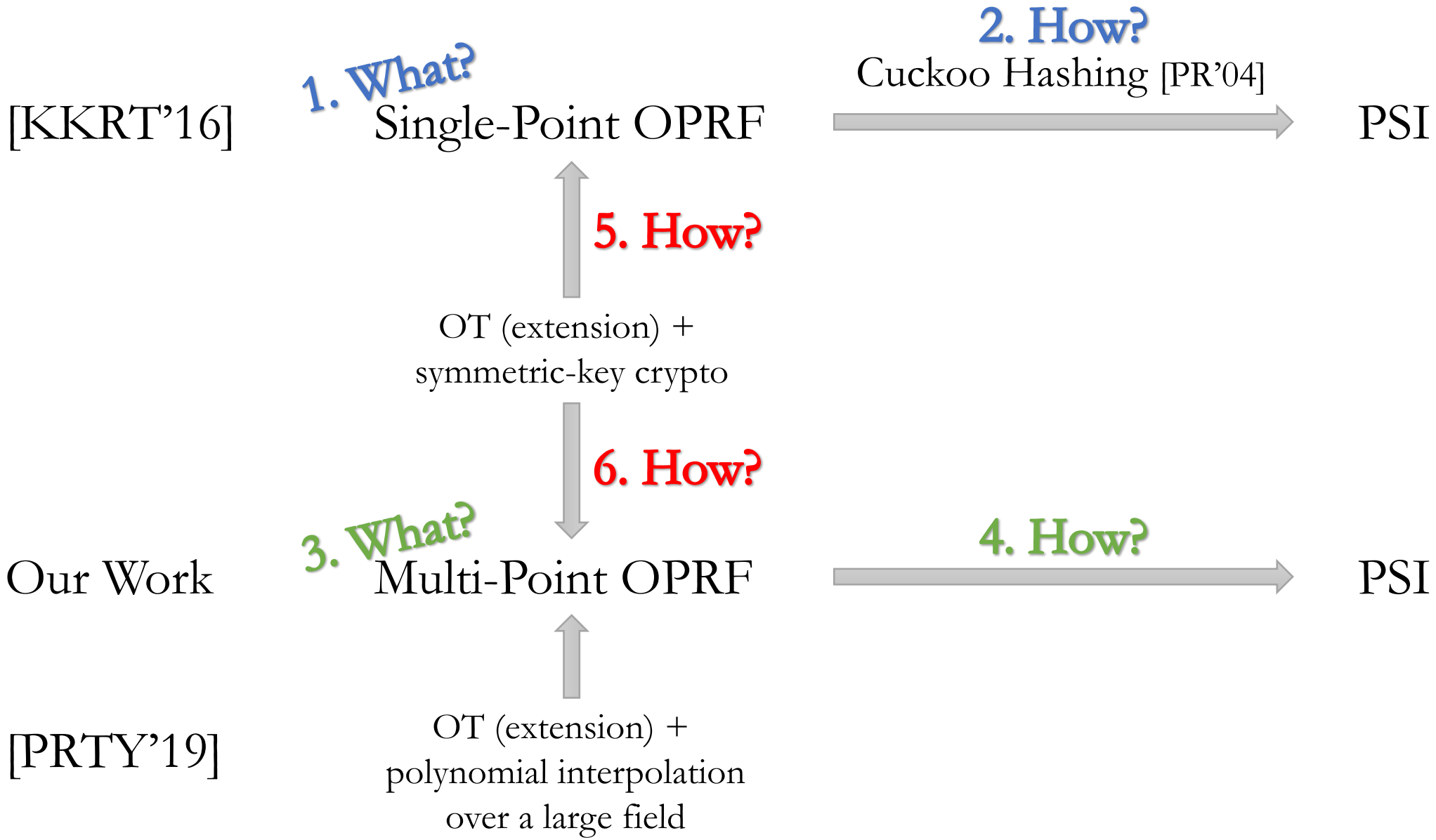
$F_k(y_i)$:

y_i → $f(y_i)$

0	1	0	1	0
1	1	1	1	0
0	0	0	1	1
0	0	0	1	0

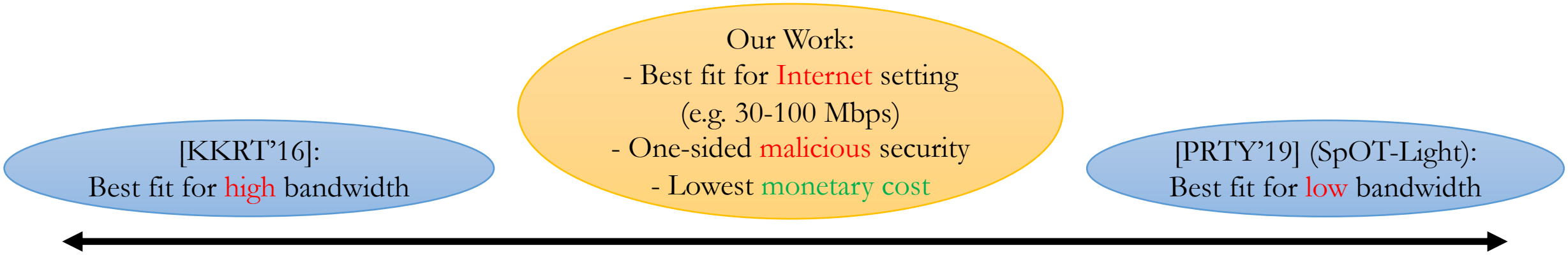
$R_1 \in \{0,1\}^{n \times \lambda}$

$H($ $)$


2. How?
Cuckoo Hashing [PR'04]
[KKRT'16]
1. What?
Single-Point OPRF
PSI
5. How?
OT (extension) +
symmetric-key crypto
6. How?
3. What?
4. How?
Our Work
Multi-Point OPRF
PSI
[PRTY'19]
OT (extension) +
polynomial interpolation
over a large field

Open Problems

- Best computation & communication?
- Malicious security against Bob?



Thank you!