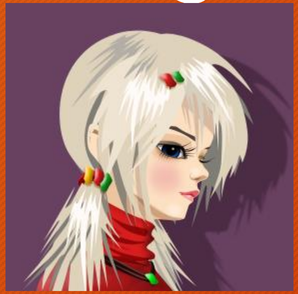
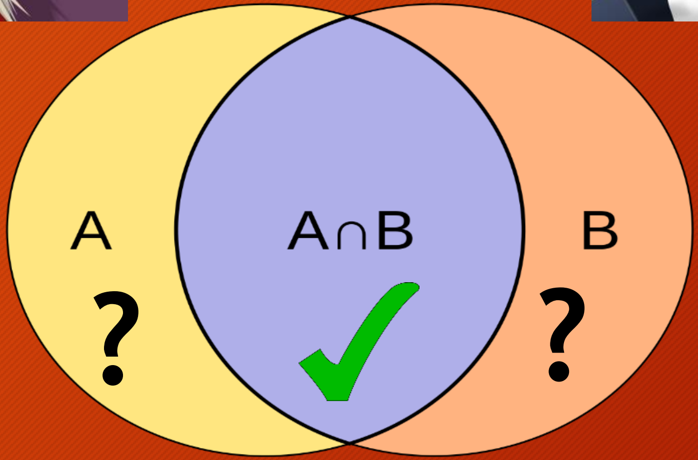
# *Are you the one to share? Secret Transfer with Access Structure*

Yongjun Zhao, Sherman S.M. Chow
Department of Information Engineering
The Chinese University of Hong Kong, Hong Kong

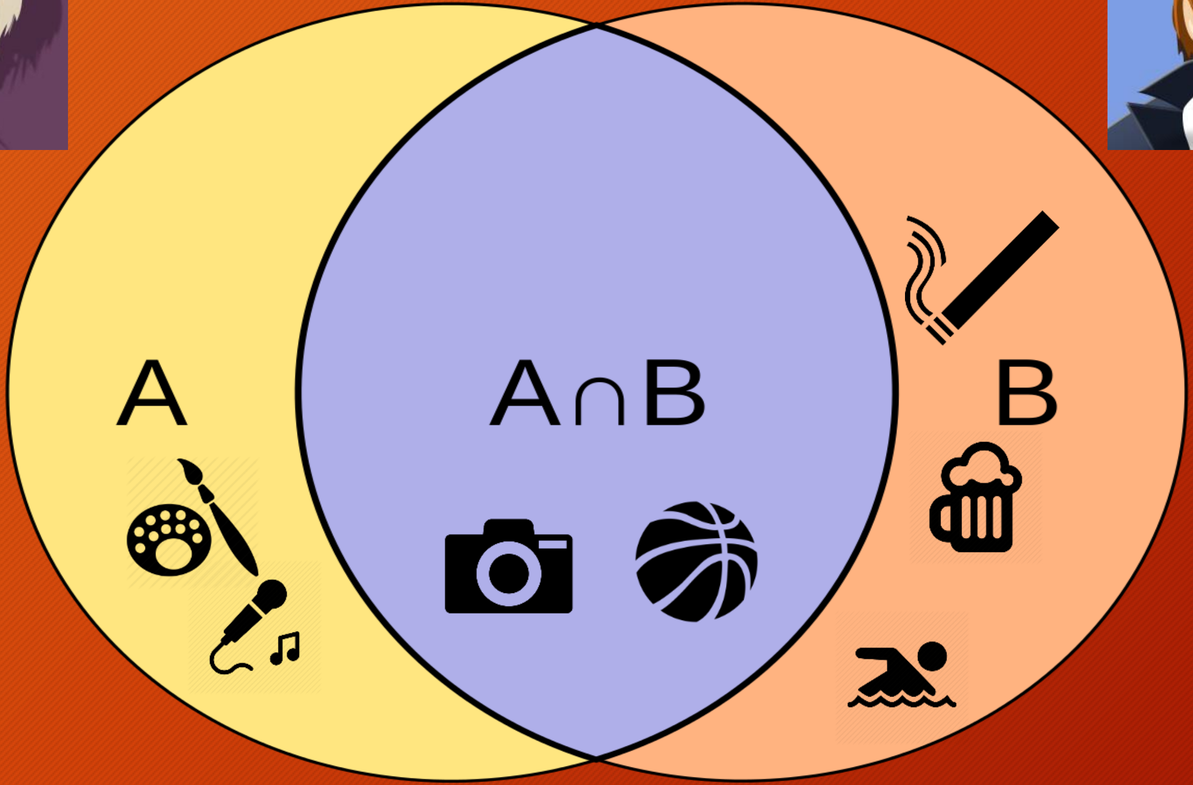
# Private Set Intersection (PSI)

- Compute the intersection $A \cap B$
- without revealing elements $\notin A \cap B$

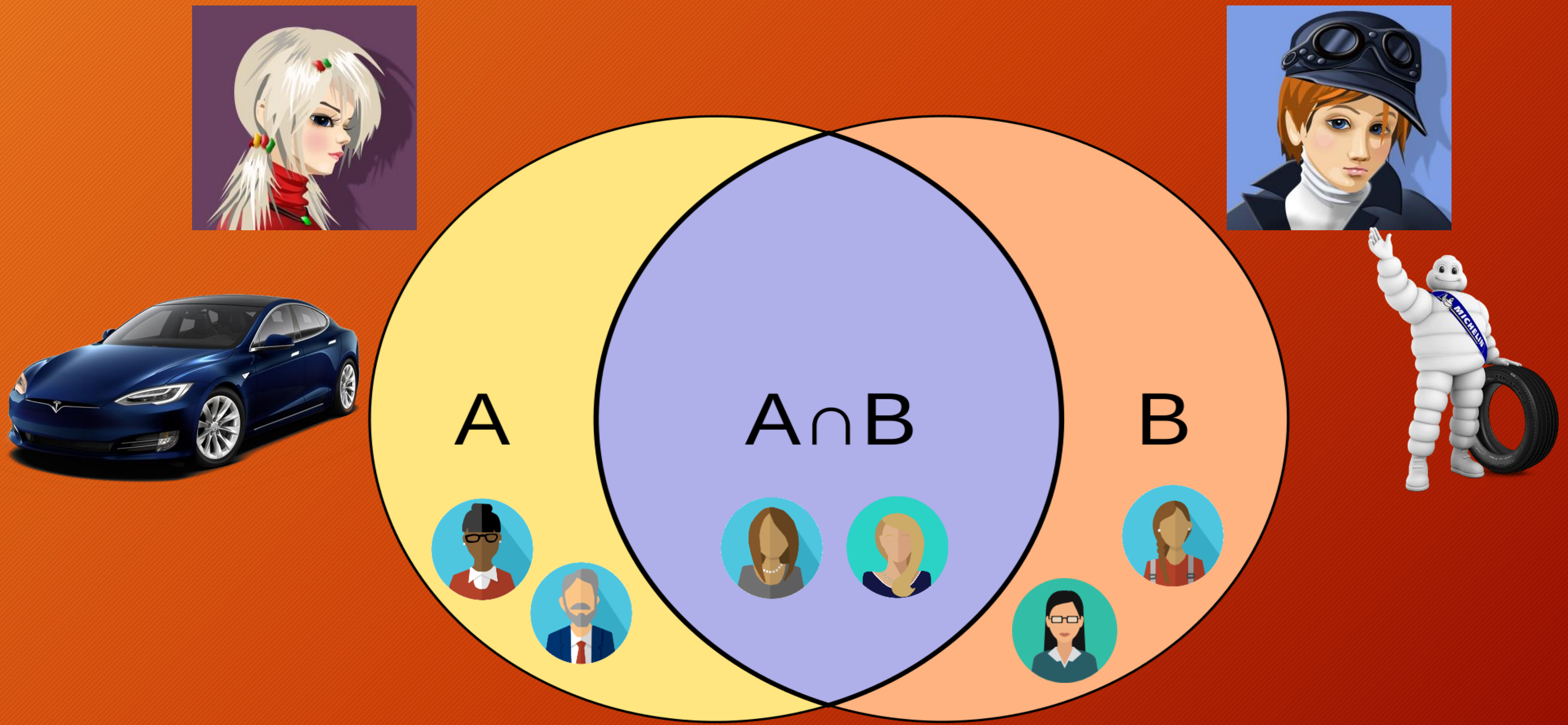




# Applications of PSI: Common Interests



# Applications of PSI: Common Customers



# Classical Definition for PSI

- $\mathcal{F}_{PSI}$: $(X, Y) \to (X \cap Y, \perp)$
- Well established notion in crypto and security communities

client  server 

| | | |
|---|---|---|
| Input: | $X = \{x_1, \dots, x_n\}$ | $Y = \{y_1, \dots, y_m\}$ |
| Output: | $X \cap Y$ | $\perp$ |

- Other variants: fair PSI (both parties obtain $X \cap Y$), multi-party PSI (>2 participants), etc.

# Classical Definition for PSI (limitation)

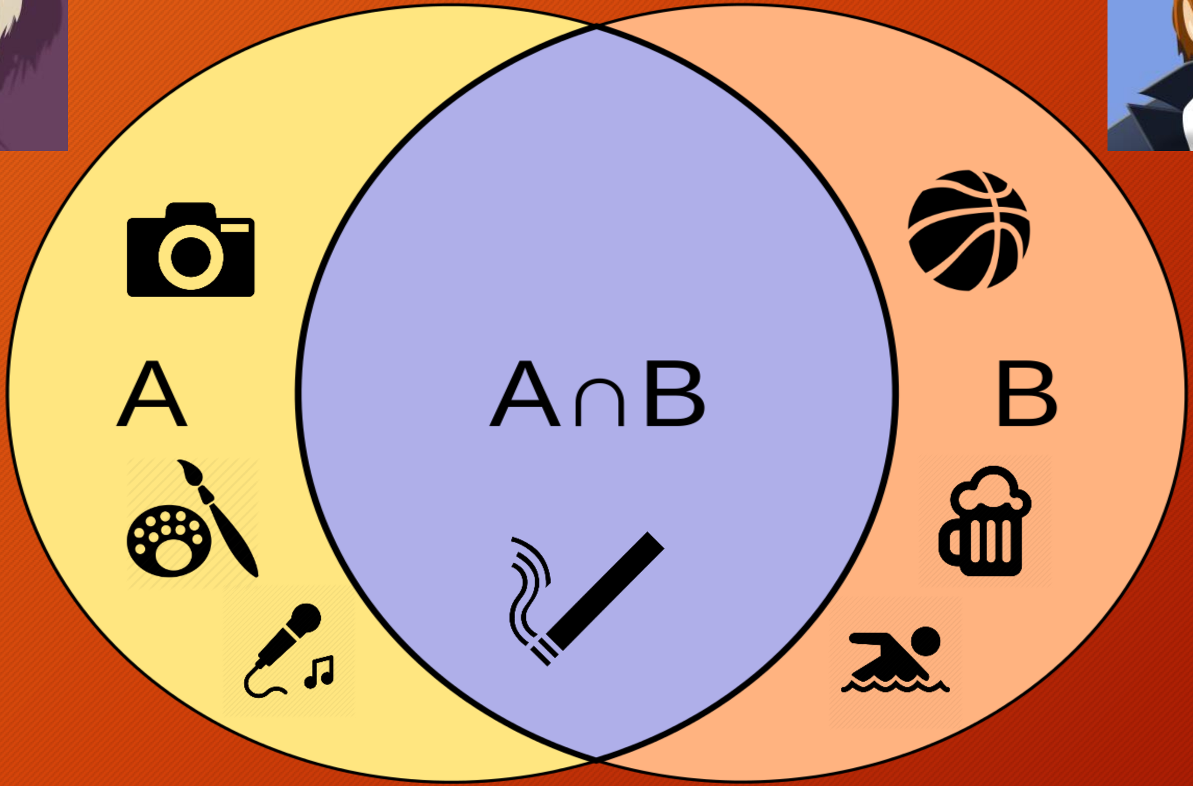
- $\mathcal{F}_{PSI}$: $(X, Y) \rightarrow (X \cap Y, \perp)$

client  server 

| | client | server |
|---|---|---|
| Input: | $X = \{x_1, \ldots, x_n\}$ | $Y = \{y_1, \ldots, y_m\}$ |
| Output: | $X \cap Y$ | $\perp$ |

- One party ALAWYS learns the outcome

# They do not really match that well



# Classical Definition (limitation)

- Traditional PSI always reveals the intersection
- Intersection set itself could be:
  - Sensitive: threat information
  - Commercial asset: customer list
  - Personal info: friend list, hobbies, preferences
- Intersection should *only* be revealed when <u>necessary</u> (i.e., the interaction satisfying some policy $P(\cdot)$)
  - e.g., the size exceeds some threshold number

# More “Privacy-Friendly” PSI

- Our new notion: PSI with (monotone) *access structure*
  - Reveal $A \cap B$ only if $P(A \cap B) = 1$
- Special cases:
  - (over) threshold PSI $P(A \cap B) = \begin{cases} 1 \text{ if } |A \cap B| \geq t \\ 0 \text{ if } |A \cap B| < t \end{cases}$

- Applications:
  - Private match-making
  - Auditing leakage in information sharing
    - Intersection of threat information / suspect lists / customer list

# Concrete Construction

- We construct PSI with access structure in a modular way
- Roadmap:

OTSA → STAS → PSI w/ AS

| OTSA | STAS | PSI w/ AS |
|---|---|---|
| Oblivious Transfer for a Sparse Array | Secret Transfer with Access Structure | PSI with Access Structure |

# Oblivious Transfer for a Sparse Array

- Roadmap:

OTSA → STAS → PSI w/ AS

Oblivious Transfer for a Sparse Array

Secret Transfer with Access Structure

PSI with Access Structure

# Oblivious Transfer for a Sparse Array (OTSA)

- $\mathcal{F}_{OTSA}$:$(x, y) \rightarrow (D, \perp)$

Input: $x = \{x_1, \ldots, x_n\}$ $y = \{(y_1, d_1), \ldots, (y_m, d_m)\}$

Output: $D = \{d_i | y_i \in \{x_1, x_2, \cdots, x_n\}\}$ $\perp$

- Generalizing standard $n$-out-of-$m$ OT:
  - $\{x_1, \ldots, x_n\} \nsubseteq \{y_1, \ldots, y_m\}$
  - $\{x_1, \ldots, x_n\} \cap \{y_1, \ldots, y_m\}$ is hidden from receiver

# Oblivious Polynomial Evaluation (OPE)

- Encode the set $\{x_1, \dots, x_n\}$ as polynomial:
  $$p = (x - x_1)(x - x_2) \cdots (x - x_n) = a_0 + a_1 x + \cdots + a_n x^n$$
- Observation: $y_i \in X \Leftrightarrow p(y_i) = 0$
- Given encrypted coefficients $a_0, a_1, \dots, a_n$ of a polynomial $p$
- We can evaluate its value at $x$ via homomorphic encryption:

$$Enc_{pk}\big(p(x)\big) = Enc_{pk}(a_0 + a_1 x + \cdots + a_n x^n)$$
$$= Enc_{pk}(a_0) \oplus \big(Enc_{pk}(a_1) \otimes x\big) \oplus \cdots \oplus \big(Enc_{pk}(a_n) \otimes x^n\big)$$

# OTSA from Oblivious Polynomial Evaluation

$pk, Enc_{pk}(a_0), \dots, Enc_{pk}(a_n)$

$\{z_1, \dots, z_m\}$ (permuted)

$z_i = Enc_{pk}(r_i \cdot p(y_i) + d_i)$

$(pk, sk)$

$\{x_1, \dots, x_n\}$

$\{y_1, \dots, y_m\}$

$\{d_1, \dots, d_m\}$

if $y_i \in \{x_1, \dots, x_n\}$ $z_i$ will be decrypted to $d_i$

if $y_i \notin \{x_1, \dots, x_n\}$ $z_i$ will be decrypted to random

# Construction of OTSA

$pk, Enc_{pk}(a_0), \dots, Enc_{pk}(a_n)$ $\longrightarrow$

$\longleftarrow$ $z_1, \dots, z_m$

$z_i = Enc_{pk}(r_i \cdot p(y_i) + d_i)$

- Honest-but-curious model
  - extended to malicious model using zero-knowledge proofs (details in the paper)
- Computational complexity: $O(mn)$ (worse than $O(n \log n)$ via generic approach)
- $O(n)$ construction (honest-but-curious) in the paper
  - based on garbled Bloom filter [Dong-Chen@CCS'13]

# PSI with Access Structure

- Roadmap:

OTSA → STAS → PSI w/ AS

Oblivious Transfer for a Sparse Array

Secret Transfer with Access Structure

PSI with Access Structure

# Secret Sharing

- Split a secret $s$ into shares
- $s$ can be reconstructed only if "qualified" subset of shares are combined

$$\text{SecretShare}(s) \rightarrow \{s_1, s_2, \dots, s_n\}$$

$$\text{Reconstruct}(s_{i_1}, s_{i_2}, \dots, s_{i_k}) \rightarrow s \text{ or } \perp$$

- Example:

access structure:

$s_1$ AND $\{s_2$ OR $s_3\}$ AND $s_4$ AND $s_5$

"qualified" subsets:

$\{s_1, s_2, s_4, s_5\}$

$\{s_1, s_3, s_4, s_5\}$

$\{s_1, s_2, s_3, s_4, s_5\}$

# Secret Transfer with Access Structure

- $\mathcal{F}_{STAS}$:

| |  |  |
|---|---|---|
| Input: | $X = \{x_1, \dots, x_n\}$ | $s,\ Y = \{y_1, \dots, y_m\}$ |
| Output: | $\lvert X \cap Y \rvert$ and<br>$s$ iff $P(X \cap Y) = 1$ | $\perp$ |

# OTSA + Secret Sharing = STAS

$pk, Enc_{pk}(a_0), \dots, Enc_{pk}(a_n) \longrightarrow$

$\longleftarrow z_1, \dots, z_m$

SecretShare($s$) $\rightarrow \{s_1, s_2, \dots, s_m\}$

$z_i = Enc_{pk}(r_i \cdot p_X(y_i) + s_i)$

$(pk, sk)$

$X = \{x_1, \dots, x_n\}$

$Y = \{y_1, \dots, y_m\}$

$s$

if $y_i \in X$ $z_i$ will be decrypted to $s_i$

if $y_i \notin X$ $z_i$ will be decrypted to random

# OTSA + Secret Sharing = STAS

$pk, Enc_{pk}(a_0), \ldots, Enc_{pk}(a_n)$

$z_1, \ldots, z_m$

$\text{SecretShare}(s) \rightarrow \{s_1, s_2, \ldots, s_m\}$

$z_i = Enc_{pk}(r_i \cdot p_X(y_i) + s_i)$

$(pk, sk)$
$X = \{x_1, \ldots, x_n\}$

$Y = \{y_1, \ldots, y_m\}$
$s$

If $X \cap Y$ satisfies the access structure
The receiver can reconstruct the secret $s$ !

# PSI with Access Structure

- Roadmap:

PSI w/ DT → STAS → PSI w/ AS

Oblivious Transfer for a Sparse Array

Secret Transfer with Access Structure

PSI with Access Structure

# PSI with Access Structure from STAS


$X = \{x_1, \dots, x_n\}$

$Y = \{y_1, \dots, y_m\}$ and $s$

The receiver can reconstruct the secret $s$
if and only if $X \cap Y$ satisfies the access structure

# STAS + PSI = PSI with Access Structure

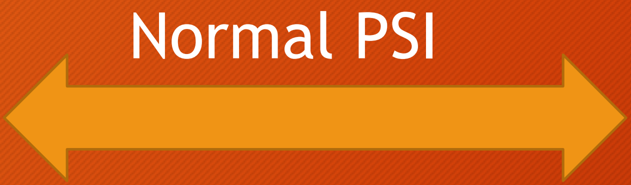

$X' = \{x_1||s, \dots, x_n||s\}$

$Y' = \{y_1||s, \dots, y_m||s\}$

If $X \cap Y$ satisfies the access structure

The receiver can learn $X' \cap Y'$, which is essentially $X \cap Y$

# PSI with Access Structure




$X' = \{x_1||s', \ldots, x_n||s'\}$

$Y' = \{y_1||s, \ldots, y_m||s\}$

If $X \cap Y$ does not satisfies the access structure
The receiver can learn $X' \cap Y'$, which is an empty set

# Concluding Remarks

- We introduce the notions of
  - Oblivious Transfer with Spare Array (OTSA)
  - Secret Transfer with Access Structure (STAS)
  - PSI with Access Structure
- We then construct
  - Two OTSA schemes (from OPE / garbled Bloom filter)
  - OTSA + Secret Sharing = STAS
  - STAS + PSI = PSI with Access Structure
- Future work 1: can we hide $|X \cap Y|$ in STAS?
- Future work 2: can we support non-monotone access structure?

Under submission


- {zy113, sherman}@ie.cuhk.edu.hk