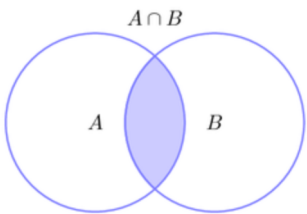
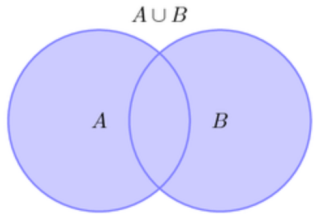
Private Set Operations





Yu Chen
Shandong University

Outline

Secure Multi-party Computation (MPC)

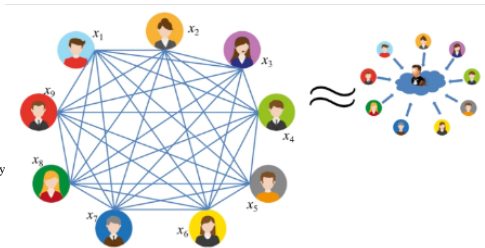
[Yao82]: Protocols for Secure Computations

ANDREW CHI-CHIH YAO

China – 2000

CITATION

In recognition of his fundamental contributions to the theory of computation, including the complexity-based theory of pseudorandom number generation, cryptography and communication complexity.



MPC enable a group of independent data owners who do not trust each other or any common third party

- jointly compute a function that depends on all of their private inputs
- without learn anything else beyond output and its own input

Short History of MPC

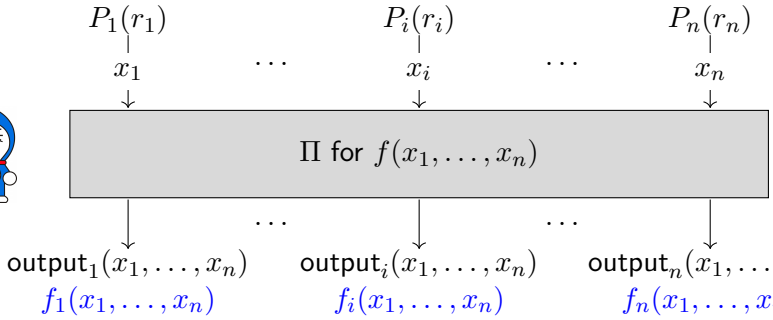
General-purpose MPC (a.k.a. can compute arbitraty function) is possible!
This makes MPC extremely powerful

Celebrated Paradigms of Generic MPC

Approach	Round Complexity	Adversarial Behavior	Party	Corruption Threshold	Protocol	Technique	Circuits
Garbled Circuit	$O(1)$	Semi-honest	$n = 2$	—	Yao [Yao86]	standard GC	Boolean
			$n \geq 2$	$t \geq n/2$	BMR [BMR90]	Distributed Garbling	
		Malicious	$n = 2$	—	LP [LP07], LEGO [NO09]	Cut-and-Choose	
					WRK [WRK17]	IT-MAC	
			$n \geq 2$	$t \geq n/2$	CKMZ [CKMZ14]	Cut-and-Choose	
					YWZ [YWZ20]	IT-MAC	
Secret Sharing	$O(d)$	Semi-honest	$n \geq 2$	$t \geq n/2$	GMW [GMW87]	Additive SS	Arithmetic
				$t < n/2$	BGW [BGW88]	Shamir SS	
		Malicious	$n \geq 2$	$t \geq n/2$	GMW [GMW87]	ZKP	
					SPDZ [DPSZ12]	IT-MAC	
				$t < n/2$	LN [LN17]	Triple Verification	
					CGHIKLN [CGH+18]	Dual Execution	
					BGIN [BGIN20]	Distributed ZKP	

d is the depth of C, typically $\log |C|$.

MPC with Semi-Honest Security



P_i's view includes
- private input
- random tape
- incoming messages

MPC with Semi-Honest Security

$P_1(r_1)$ $\quad \cdots \quad$ $P_i(r_i)$ $\quad \cdots \quad$ $P_n(r_n)$

$x_1 \quad \cdots \quad x_i \quad \cdots \quad x_n$

Π for $f(x_1, \ldots, x_n)$

$\cdots \quad \cdots$

$\mathsf{output}_1(x_1, \ldots, x_n) \quad \mathsf{output}_i(x_1, \ldots, x_n) \quad \mathsf{output}_n(x_1, \ldots, x_n)$

$f_1(x_1, \ldots, x_n) \quad f_i(x_1, \ldots, x_n) \quad f_n(x_1, \ldots, x_n)$

P_i's view includes
private input
random tape
incoming messages

- all P_i are semi-honest (honest but curious)
- P_i learns no more information other than his output and private input

Definition 1 (Semi-honest Security)

Π securely realizes probabilistic f in the presence of semi-honest adversaries if there exists a PPT simulator Sim such that for all inputs $x_1, \ldots, x_n$ and all $i \in [n]$:

$(\mathsf{View}_{P_i}(x_1, \ldots, x_n), \mathsf{output}(x_1, \ldots, x_n)) \approx_{c,s} (\mathsf{Sim}(i, x_i, f_i(x_1, \ldots, x_n)), f(x_1, \ldots, x_n))$

MPC with Semi-Honest Security

$P_1(r_1)$ $\quad \cdots \quad$ $P_i(r_i)$ $\quad \cdots \quad$ $P_n(r_n)$

$x_1 \quad \cdots \quad x_i \quad \cdots \quad x_n$

Π for $f(x_1, \ldots, x_n)$

$\cdots \quad \cdots$

$\mathsf{output}_1(x_1, \ldots, x_n) \quad \mathsf{output}_i(x_1, \ldots, x_n) \quad \mathsf{output}_n(x_1, \ldots, x_n)$

$f_1(x_1, \ldots, x_n) \quad f_i(x_1, \ldots, x_n) \quad f_n(x_1, \ldots, x_n)$

P_i's view includes
private input
random tape
incoming messages

- all P_i are semi-honest (honest but curious)
- P_i learns no more information other than his output and private input

Definition 1 (Semi-honest Security)

Π securely realizes deterministic f in the presence of semi-honest adversaries if there exists a PPT simulator Sim such that for all inputs $x_1, \ldots, x_n$ and all $i \in [n]$:

$$\mathsf{View}_{P_i}(x_1, \ldots, x_n) \approx_{c,s} \mathsf{Sim}(i, x_i, f_i(x_1, \ldots, x_n))$$

Short History of MPC

MPC was primarily of only theoretical interest for the first twenty years.

- After 2000s, algorithmic improvements and computing costs reached a point where generic MPC is practical for real-world applications.

But, generic MPC is still relatively heavy and thus not very fast!

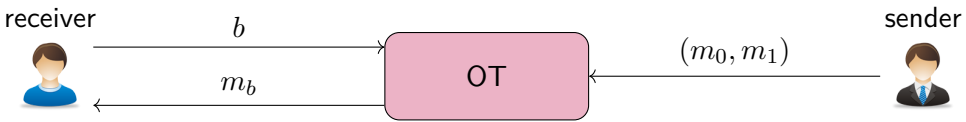
One important sub-area of MPC focuses on specific functionalities

- For specific functionalities, there maybe custom protocols that are much more efficient than the best generic protocols.
- Specific functionalities can be interesting in their own right, but also can be natural building blocks for use in other applications.


这个就叫专业

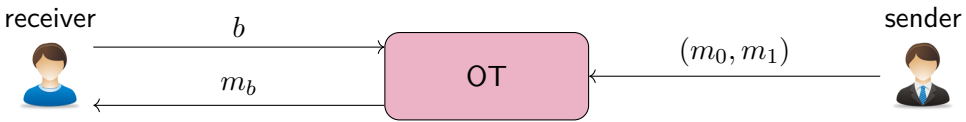
Oblivious Transfer

1-out-of 2 OT [Rab05] enables the receiver learns only one messages from sender, while sender learns nothing.



Oblivious Transfer

1-out-of 2 OT [Rab05] enables the receiver learns only one messages from sender, while sender learns nothing.

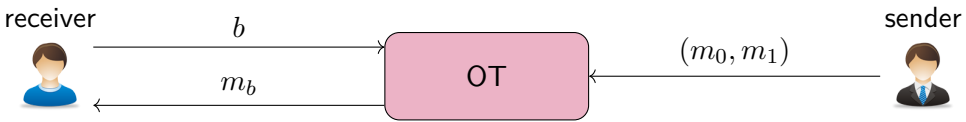


OT is complete for MPC [Kil88].

- Private-information retrieval (PIR) is weaker than OT: it only cares privacy of receiver

Oblivious Transfer

1-out-of 2 OT [Rab05] enables the receiver learns only one messages from sender, while sender learns nothing.



OT is complete for MPC [Kil88].

- Private-information retrieval (PIR) is weaker than OT: it only cares privacy of receiver

OT does not belong to Minicrypt $\rightsquigarrow$ expensive public-key operations are unavoidable, while real applications need a large number of OT

- [IKNP03] proposed Ishai-Kilian-Nissim-Petrank OT extension: generate many OT efficiently from $O(\kappa)$ number of base OT $\Rightarrow$ OTe is cheap

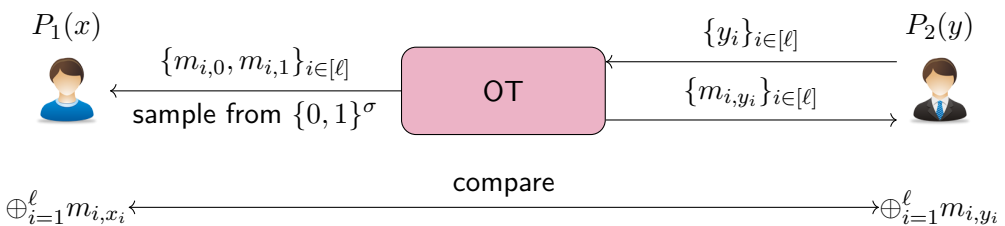
Private Equality Test Protocol

PEQT [PSSZ15] enables P_1 and P_2 check if their ℓ-bits elements x and y are equal.

Private Equality Test Protocol

PEQT [PSSZ15] enables P_1 and P_2 check if their ℓ-bits elements x and y are equal.

[PSSZ15] showed how to build PEQT by invoking 1-out-of-2 random OT ℓ times



Oblivious Pseudorandom Functions

OPRF [FIPR05] enables server obtain a key k and client evaluate obliviously.

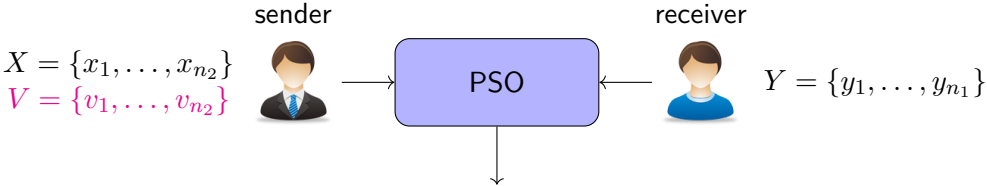
Oblivious Pseudorandom Functions

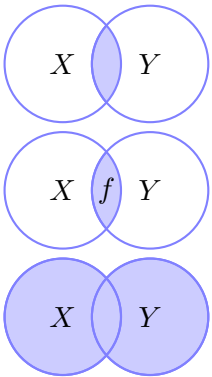
OPRF [FIPR05] enables server obtain a key k and client evaluate obliviously.

OPRF is a powerful tool in MPC (see [CHL22] for a good survey)

- many variants: batch/programmable/permuted/distributed OPRF
- fast construction from OT or VOLE

Specific Functionality: Private Set Operations (high frequency and high value)

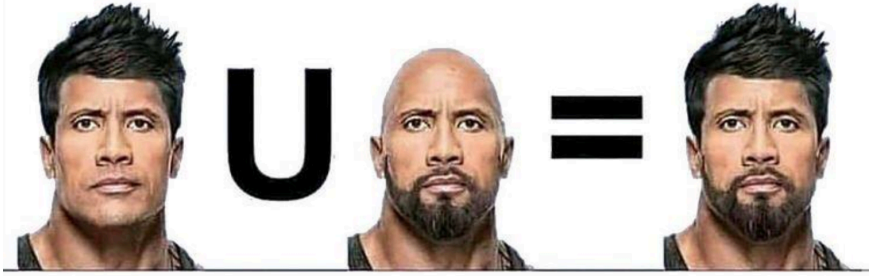




$$\text{PSI} = X \cap Y$$

$$\text{PCSI} = \begin{cases} |X \cap Y| & \text{cardinality} \\ |X \cap Y|, \sum_{x_i \in X \cap Y} v_i & \text{cardinality-sum} \\ f(X \cap Y) & \text{general computation} \end{cases}$$

$$\text{PSU} = X \cup Y$$



Applications of PSI

Contact discovery (when signing up an App)

- X: address book in my phone
- Y: App user database

Private scheduling

- X: avaiable timeslots on my calendar
- Y: avaiable timeslots on your calendar

Credit risk profiling

- X: bank's credit risk watchlist
- Y: prospective borrowers

Password checkup

- X: Google's database of breached passwords
- Y: client's passwords

Applications of PSI-card-sum and PSU

Ad conversion rate (widely used in Microsoft and Google)

- X: users who saw the advertisement
- Y: customers who bought the product

IP blacklist and vulnerability data aggregation

- X: blacklist in organization A
- Y: blacklist in organization B

Private DB supporting full join

- X: data from table A
- Y: data from table B

SOTA of PSI

PSI has been extensively studied in the last four decades

According to the techniques

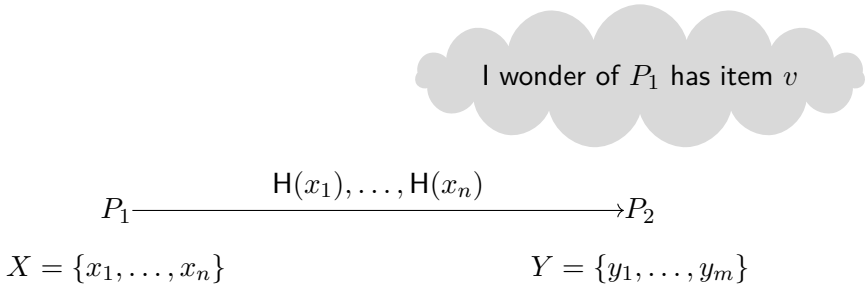
- symmetric-key: OPRF-based [KKRT16, CM20, RR22]
- public-key: communication-efficient DH-PSI [Mea86]

According to the scenarios

- balanced setting: [KKRT16, CM20, RR22] achieves linear complexity
- unbalanced setting: [CLR17, CHLR18, CMdG+21] achieves sub-linear complexity of large set

the SOTA [RR22] s almost as efficient as insecure hash-based protocol
million size set: [0.16s, 31 Mb]

Naive Insecure Hash-based Protocol

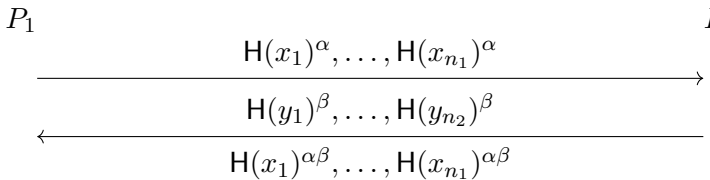


INSECURE: $\mathsf{H}(X)$ reveal too much information $\rightsquigarrow$ P_2 can test any $v? \in X$ offline

- Especially problematic if items have low entropy (e.g., phone numbers)

Recap of Classic DH-PSI

$X = \{x_1, \ldots, x_{n_1}\}$ $\qquad$ $Y = \{y_1, \ldots, y_{n_2}\}$



compute $X \cup Y$ by comparing
$\mathsf{H}(x_i)^{\alpha\beta} \in \{\mathsf{H}(y_i)^{\beta\alpha}\}_{i\in[n_2]}$

Idea:

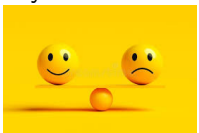
- If $x_i \in Y$, then $\mathsf{H}(x_i)^{\alpha\beta} = \mathsf{H}(y_i)^{\beta\alpha}$ for some $y_j \in Y$
- If $x_i \notin Y$, messages are independently random by modeling H as random oracle

Exercise

Prove the classic DH-PSI protocol in the semi-honest security model.

SOTA of PSO

In sharp contrast, the study of PCSI and PSU are not satisfying.



PCSI

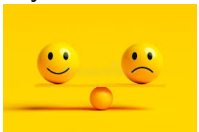
- [HFH99, IKN+20, PSTY19] achieve linear complexity

concretely $20\times$ slower in timing and $30\times$ more communication than PSI

Somewhat counter-intuitive, less is harder!

SOTA of PSO

In sharp contrast, the study of PCSI and PSU are not satisfying.



PCSI

- [HFH99, IKN+20, PSTY19] achieve linear complexity

concretely $20\times$ slower in timing and $30\times$ more communication than PSI

Somewhat counter-intuitive, less is harder!

PSU

- [KS05, Fri07, HN10, KRTW19, JSZ+22] have superlinear complexity
- [DC17] achieve linear complexity, but not strict (communication or computation complexity additionally depends on statistical parameter $\lambda \approx 40$)

concretely $20\times$ slower in timing and $25\times$ more communication than PSI

Need to fetch information belong to other parties.

Motivation

Different approaches are used for different private set operations $\rightsquigarrow$ require much more engineering effort and maintaining cost

- Goal: a unified framework of PSO

[1] [GMR+21] presented a PSO framework from permuted characteristic. However, its oblivious shuffle functionality is not necessary for PSO, and incurs superlinear complexity.

Motivation

Different approaches are used for different private set operations $\rightsquigarrow$ require much more engineering effort and maintaining cost

- Goal: a unified framework of PSO

There exists huge efficiency gap between PSI and other PSO protocols

- Goal: efficient instantiations to close the gap[1]

[1] [GMR+21] presented a PSO framework from permuted characteristic. However, its oblivious shuffle functionality is not necessary for PSO, and incurs superlinear complexity.

Motivation

Different approaches are used for different private set operations $\rightsquigarrow$ require much more engineering effort and maintaining cost

- Goal: a unified framework of PSO

There exists huge efficiency gap between PSI and other PSO protocols

- Goal: efficient instantiations to close the gap[1]

After $\approx$ 40 years, DH-PSI [Mea86] is still the most easily understood and implemented one among numerous PSI protocols. Surprisingly, no counterpart is known in the PSU setting yet. Existing protocols are very complicated.

- Goal: build DDH-based PSU protocol as simple as DH-PSI

[1] [GMR+21] presented a PSO framework from permuted characteristic. However, its oblivious shuffle functionality is not necessary for PSO, and incurs superlinear complexity.

Questions in Mind

1. *Is there a central building block that enables a unified framework for PSO?*
2. *How to give instantiations with optimal asymptotic complexity and good concrete efficiency?*
3. *Can the DDH assumption strike back with efficient PSU protocol?*

Reference I


Elette Boyle, Niv Gilboa, Yuval Ishai, and Ariel Nof.
Efficient fully secure computation via distributed zero-knowledge proofs.
In *Advances in Cryptology - ASIACRYPT 2020*, volume 12493 of *Lecture Notes in Computer Science*, pages 244–276. Springer, 2020.


Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson.
Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract).
In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 1988*, pages 1–10. ACM, 1988.


Donald Beaver, Silvio Micali, and Phillip Rogaway.
The round complexity of secure protocols (extended abstract).
In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, 1990*, pages 503–513. ACM, 1990.

Reference II


Koji Chida, Daniel Genkin, Koki Hamada, Dai Ikarashi, Ryo Kikuchi, Yehuda Lindell, and Ariel Nof.
Fast large-scale honest-majority mpc for malicious adversaries.
In *Advances in Cryptology - CRYPTO 2018*, volume 10993 of *Lecture Notes in Computer Science*, pages 34–64. Springer, 2018.


Sílvia Casacuberta, Julia Hesse, and Anja Lehmann.
Sok: Oblivious pseudorandom functions.
In *7th IEEE European Symposium on Security and Privacy, EuroS&P 2022*, pages 625–646. IEEE, 2022.


Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal.
Labeled PSI from fully homomorphic encryption with malicious security.
In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, pages 1223–1237. ACM, 2018.

Reference III


Seung Geol Choi, Jonathan Katz, Alex J Malozemoff, and Vassilis Zikas.
Efficient three-party computation from cut-and-choose.
In *Advances in Cryptology - CRYPTO 2014*, volume 8617 of *Lecture Notes in Computer Science*, pages 513–530. Springer, 2014.


Hao Chen, Kim Laine, and Peter Rindal.
Fast private set intersection from homomorphic encryption.
In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017*, pages 1243–1255. ACM, 2017.


Melissa Chase and Peihan Miao.
Private set intersection in the internet setting from lightweight oblivious PRF.
In *Advances in Cryptology - CRYPTO 2020*, volume 12172 of *Lecture Notes in Computer Science*, pages 34–63. Springer, 2020.

Reference IV


Kelong Cong, Radames Cruz Moreno, Mariana Botelho da Gama, Wei Dai, Ilia Iliashenko, Kim Laine, and Michael Rosenberg.
Labeled PSI from homomorphic encryption with reduced computation and communication.
In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1135–1150. ACM, 2021.


Alex Davidson and Carlos Cid.
An efficient toolkit for computing private set operations.
In *Information Security and Privacy - 22nd Australasian Conference, ACISP 2017*, volume 10343 of *Lecture Notes in Computer Science*, pages 261–278. Springer, 2017.

Reference V


Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias.
Multiparty computation from somewhat homomorphic encryption.
In *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 643–662. Springer, 2012.


Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold.
Keyword search and oblivious pseudorandom functions.
In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 303–324. Springer, 2005.


Keith B. Frikken.
Privacy-preserving set union.
In *Applied Cryptography and Network Security, 5th International Conference, ACNS 2007*, volume 4521 of *Lecture Notes in Computer Science*, pages 237–252. Springer, 2007.

Reference VI


Gayathri Garimella, Payman Mohassel, Mike Rosulek, Saeed Sadeghian, and Jaspal Singh.
Private set operations from oblivious switching.
In *Public-Key Cryptography - PKC 2021*, volume 12711 of *Lecture Notes in Computer Science*, pages 591–617. Springer, 2021.


Oded Goldreich, Silvio Micali, and Avi Wigderson.
How to play any mental game or A completeness theorem for protocols with honest majority.
In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987*, pages 218–229. ACM, 1987.


Bernardo A. Huberman, Matthew K. Franklin, and Tad Hogg.
Enhancing privacy and trust in electronic communities.
In *Proceedings of the First ACM Conference on Electronic Commerce (EC-99)*, pages 78–86. ACM, 1999.

Reference VII


Carmit Hazay and Kobbi Nissim.
Efficient set operations in the presence of malicious adversaries.
In *Public Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 312–331. Springer, 2010.


Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung.
On deploying secure computing: Private intersection-sum-with-cardinality.
In *IEEE European Symposium on Security and Privacy, EuroS&P 2020*, pages 370–389. IEEE, 2020.


Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank.
Extending oblivious transfers efficiently.
In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer, 2003.

Reference VIII


Yanxue Jia, Shi-Feng Sun, Hong-Sheng Zhou, Jiajun Du, and Dawu Gu.
Shuffle-based private set union: Faster and more secure.
In *USENIX 2022*, 2022.


Joe Kilian.
Founding cryptography on oblivious transfer.
In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 20–31. ACM, 1988.


Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu.
Efficient batched oblivious PRF with applications to private set intersection.
In *CCS 2016*, pages 818–829. ACM, 2016.


Vladimir Kolesnikov, Mike Rosulek, Ni Trieu, and Xiao Wang.
Scalable private set union from symmetric-key techniques.
In *Advances in Cryptology - ASIACRYPT 2019*, volume 11922 of *Lecture Notes in Computer Science*, pages 636–666. Springer, 2019.

Reference IX


Lea Kissner and Dawn Xiaodong Song.
Privacy-preserving set operations.
In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 241–257. Springer, 2005.


Yehuda Lindell and Ariel Nof.
A framework for constructing fast mpc over arithmetic circuits with malicious adversaries and an honest-majority.
In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017*, pages 259–276, 2017.


Yehuda Lindell and Benny Pinkas.
An efficient protocol for secure two-party computation in the presence of malicious adversaries.
In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 52–78. Springer, 2007.

Reference X


Catherine A. Meadows.
A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party.
In *Proceedings of the 1986 IEEE Symposium on Security and Privacy*, pages 134–137. IEEE Computer Society, 1986.


Jesper Buus Nielsen and Claudio Orlandi.
Lego for two-party secure computation.
In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *Lecture Notes in Computer Science*, pages 368–386. Springer, 2009.


Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner.
Phasing: Private set intersection using permutation-based hashing.
In *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015*, pages 515–530. USENIX Association, 2015.

Reference XI


Benny Pinkas, Thomas Schneider, Oleksandr Tkachenko, and Avishay Yanai.
Efficient circuit-based PSI with linear communication.
In *Advances in Cryptology - EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 122–153. Springer, 2019.


Michael O. Rabin.
How to exchange secrets with oblivious transfer.
2005.
http://eprint.iacr.org/2005/187.


Srinivasan Raghuraman and Peter Rindal.
Blazing fast PSI from improved OKVS and subfield VOLE.
In *ACM CCS 2022*, 2022.

Reference XII


Xiao Wang, Samuel Ranellucci, and Jonathan Katz.
Authenticated garbling and efficient maliciously secure two-party computation.
In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017*, pages 21–37. ACM, 2017.


Andrew Chi-Chih Yao.
Theory and applications of trapdoor functions (extended abstract).
In *23rd Annual Symposium on Foundations of Computer Science, FOCS 1982*, pages 80–91. IEEE Computer Society, 1982.


Andrew Chi-Chih Yao.
How to generate and exchange secrets (extended abstract).
In *27th Annual Symposium on Foundations of Computer Science, FOCS 1986*, pages 162–167. IEEE Computer Society, 1986.

Reference XIII


Kang Yang, Xiao Wang, and Jiang Zhang.
More efficient mpc from improved triple generation and authenticated garbling.
In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS 2020*, pages 1627–1646, 2020.