Secure Multi-party Computation

Joseph Connor, Parker Diamond, Jordan Holland

Test Questions

1. How much data does a zero knowledge proof emit?
2. If a function f is additively homomorphic, what is f(a)+f(b) in terms of f?
3. In oblivious transfer, what does the sender learn? The receiver?

About Me - Jordan Holland

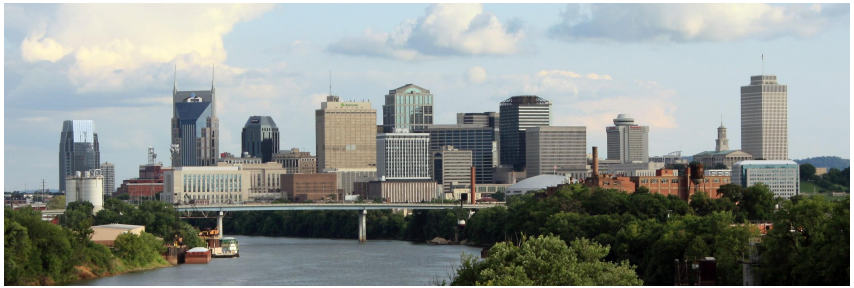
- From Hendersonville, Tennessee
- 5 Year MS-BS in Computer Science Candidate
- Interests inside of Computer Science: Security, Networks, Privacy
- Interests outside of School: Long distance backpacking, Beer
- Goals: Continuing on to a PhD program elsewhere

About Me - Jordan Holland

About Me - Joseph Connor

- From Nashville, Tennessee
- PhD student in Computer Science
- Interests inside of Computer Science: Security, Cryptography
- Interests outside of School: Hiking, Backpacking
- Goals: Industry Research

About Me - Joseph Connor



About Me - Parker Diamond

- From Chattanooga, Tennessee
- 5 Year MS-BS in Computer Science Candidate
- Interests inside of Computer Science: Security, Tor, IoT
- Interests outside of School: Lockpicking, Video Games
- Goals: Transition into Industry Research

About Me - Parker Diamond

Topics

- **Overview**
- History
- Algorithms
- Applications
- Implementations
- Open Issues
- Discussion

Overview

- Secure MPC: 2 or more parties have some *private* input
  - Want to jointly compute a *public* function of their inputs
- Zero Knowledge Proofs (ZKP)
  - Probabilistic Proofs!
  - Interactive vs Non-Interactive
  - Applications: Zcash, Ethereum

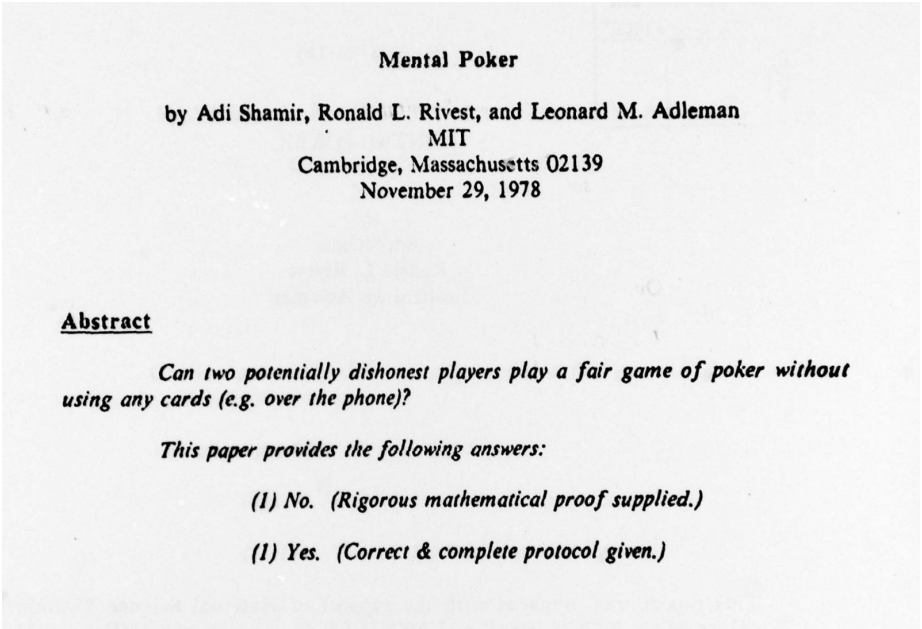
History

- 1979: "Mental Poker" (Shamir)
  - Can two players fairly deal each other private hands? No — and yes.
- 1981: Oblivious Transfer (Rabin)
  - Sender sends receiver one of two values without learning which one was sent
- 1981: Coin flipping (Blum)
  - Two parties jointly compute a random, unbiased bit
- 1982: Secure Multiparty Computation (Yao)
  - Collects and generalizes all of the above results
  - Introduces Millionaire's Problem
  - *n* parties can securely compute the output of any function!

History (Continued)

- 1985: Zero Knowledge Proofs (Goldwasser)
  - Interactive, probabilistic Proofs Introduced
  - How much knowledge is needed to prove a theorem T?
- 1986: Garbled Circuits
  - Practical two-party evaluation of general circuits (semi-honest setting)
- 1988: Non Interactive Zero Knowledge Proofs (Blum)
  - Even Less Interaction!
- 2004: Efficient Private Matching and Set Intersection (Freedman)
  - “Efficient” -- Everything is relative...
- 2017: Zero Knoweldge Proofs In Action!
  - First Zcash ZKP verified on the Ethereum network!

Mental Poker



Mental Poker

by Adi Shamir, Ronald L. Rivest, and Leonard M. Adleman
MIT
Cambridge, Massachusetts 02139
November 29, 1978

Abstract

Can two potentially dishonest players play a fair game of poker without using any cards (e.g. over the phone)?

This paper provides the following answers:

(1) No. (Rigorous mathematical proof supplied.)

(1) Yes. (Correct & complete protocol given.)

Topics

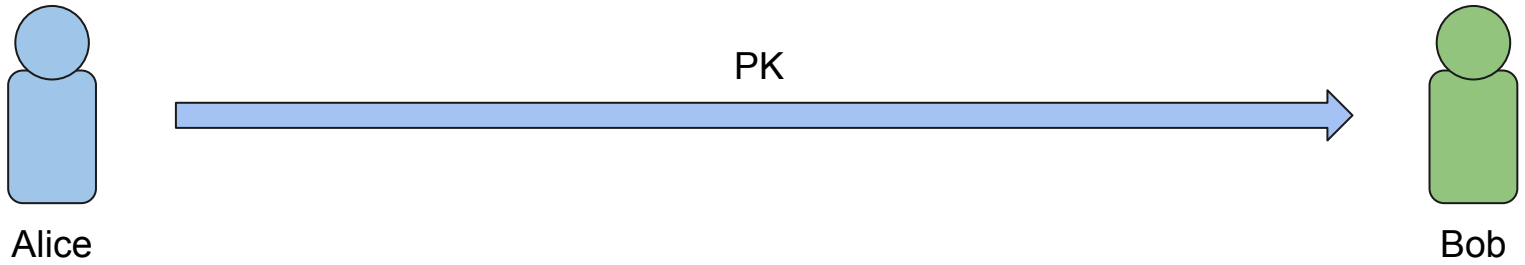
- Overview
- History
- **Algorithms**
- Applications
- Implementations
- Open Issues
- Discussion

1 of 2 Oblivious Transfer (OT)

- Alice (sender) has 2 secret values, m_0 and m_1, and wants to send Bob (receiver) exactly one
- Bob chooses b = 0 or b = 1
- Bob learns only m_b
- Alice does not learn anything new (including Bob's choice)
- Can be done with any public key system where encryption/decryption are inverses (EGL85)
    - Enc(Dec(m)) = m and Dec(Enc(m)) = m
    - RSA satisfies this property

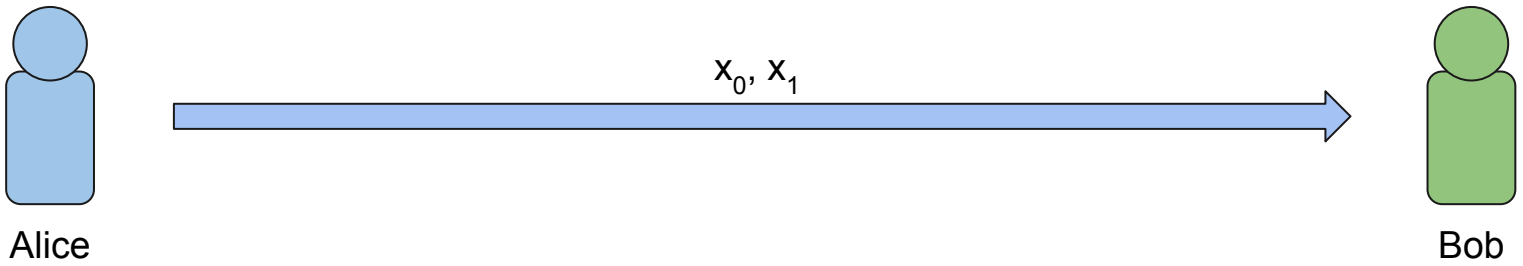
1 of 2 Oblivious Transfer

Messages: m_0, m_1
Public key PK
Secret key SK

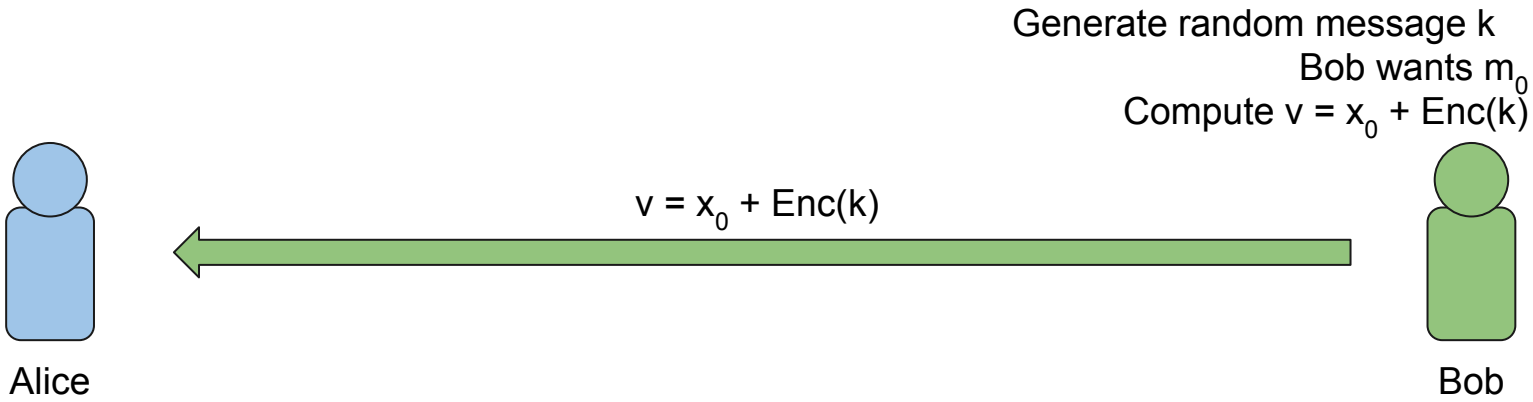


1 of 2 Oblivious Transfer

Generate random messages: x_0, x_1



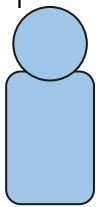
1 of 2 Oblivious Transfer



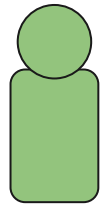
Bob can encrypt using Alice's public key.
Alice cannot tell which x Bob chose.

1 of 2 Oblivious Transfer

Compute:
$k_0 = Dec(v - x_0) = Dec(x_0 - x_0 + Enc(k)) = k$
$k_1 = Dec(v - x_1) = Dec(x_0 - x_1 + Enc(k)) =$  (random garbage)



Alice

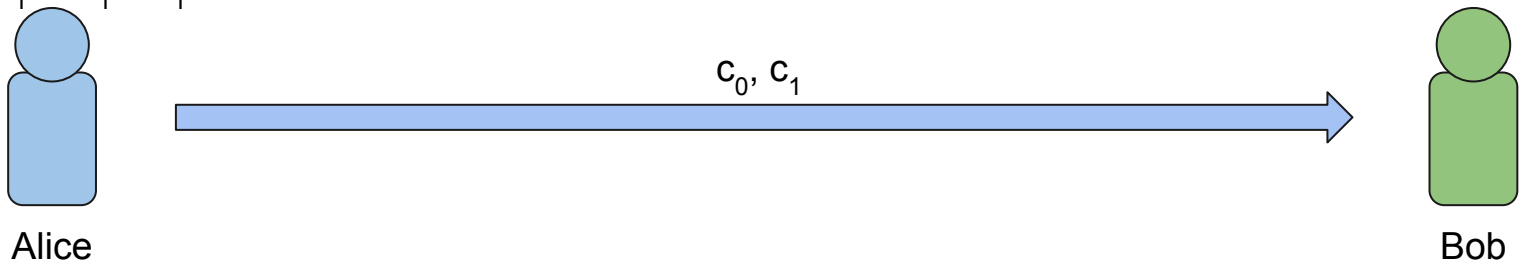


Bob

Alice can decrypt with her private key.
One of Alice's k is Bob's k, but Alice cannot tell which.

1 of 2 Oblivious Transfer

Compute:
$c_0 = m_0 + k_0$
$c_1 = m_1 + k_1$

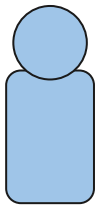


1 of 2 Oblivious Transfer

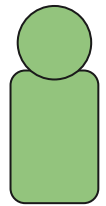
Compute:

$c_0 - k = m_0 + k - k = \mathbf{m_0}$

$c_1 - k = m_1 + k_1 - k =$ 🗑



Alice



Bob

Since $k_0 = k$, Bob can remove k_0 from c_0.
Bob cannot compute k_1, so he cannot find m_1.

OT - Generalizations and Extensions

- 1 of n OT, k of n OT
  - Both can be implemented using only 1 of 2 OT
- OT Extensions
  - Uses a few OTs plus symmetric crypto to accomplish a larger number of OTs
  - Significantly more efficient

Garbled Circuits

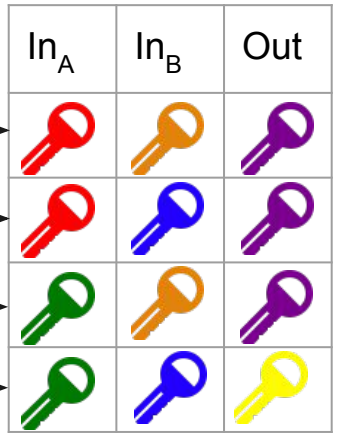
- Solution for two-party computation
- Secure against semi-honest adversaries
  - (Assumes parties will not deviate from protocol)
- Idea: encrypt inputs and outputs for any Boolean logic gate
  - Knowing two inputs allows one to decrypt the gate output
  - Compose encrypted gates to form any circuit
- Protocol:
  - Alice generates a garbled circuit and sends it to Bob, along with her encrypted inputs
  - Bob requests his encrypted inputs from Alice using OT
  - Bob evaluates the circuit and shares the result with Alice

Garbled Circuits - Encrypted AND Gate

In_A	In_B	Out
0	0	0
0	1	0
1	0	0
1	1	1

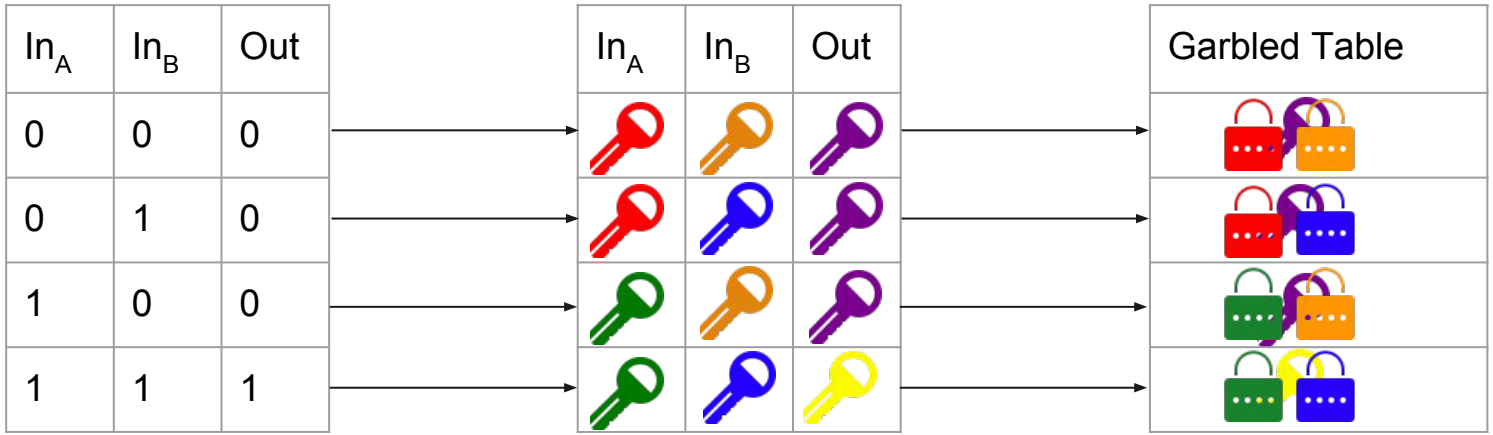
Garbled Circuits - Encrypted AND Gate

In_A	In_B	Out
0	0	0
0	1	0
1	0	0
1	1	1

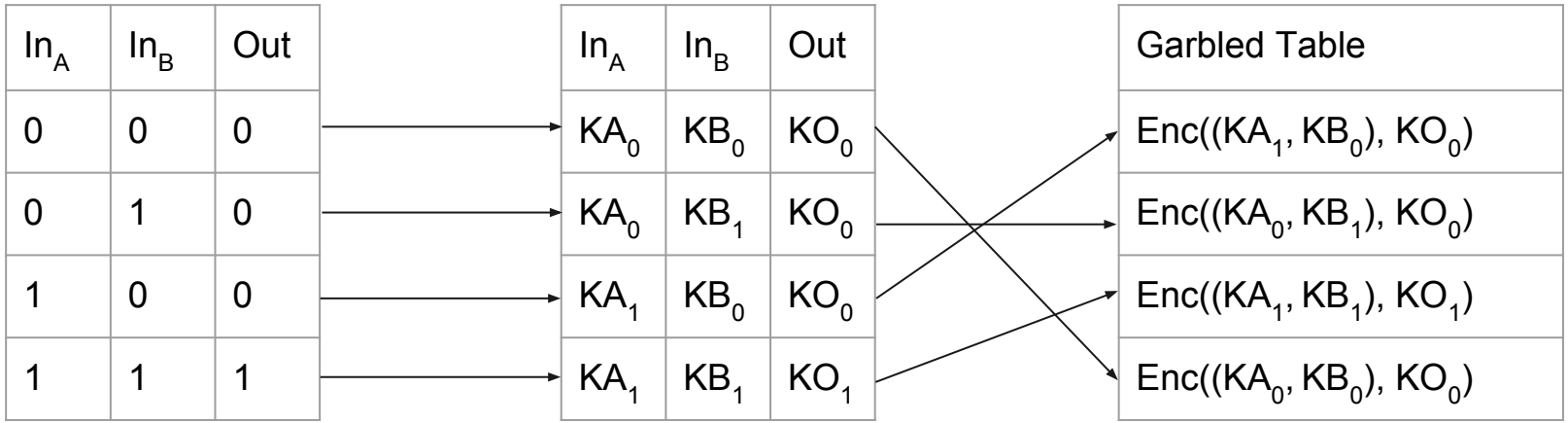


Replace 0,1 with random values (keys)

Garbled Circuits - Encrypted AND Gate



Garbled Circuits - Encrypted AND Gate



Garbled Circuits - Encrypted AND Gate

- Knowing one key for each input allows you to decrypt the garbled output
- Evaluator can try decrypting each row and check if correct
  - Key can be padded with a constant to allow checking for correct decryption
- The output of one gate can be used as input for another
  - Allows computation of any function as a circuit

Garbled Table
$Enc((KA_1, KB_0), KO_0)$
$Enc((KA_0, KB_1), KO_0)$
$Enc((KA_1, KB_1), KO_1)$
$Enc((KA_0, KB_0), KO_0)$

Garbled Circuits - Optimizations

- Point-and-permute: allows evaluator to know which row to decrypt
  - Avoids need to try decrypting all 4 rows
- Row reduction: fixes value of first row so that only three need to be sent
  - Compact circuit representation, but no faster to compute
- Free XOR:
  - Allows computation of XOR gates as bitwise XOR of inputs

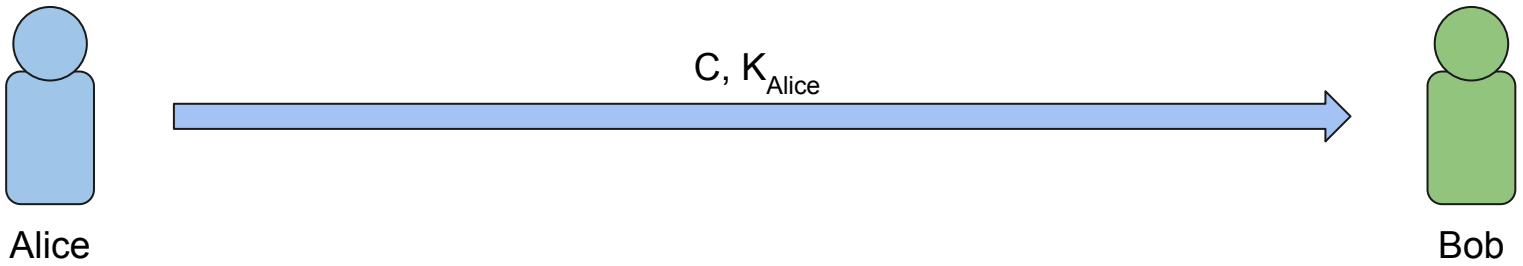
Garbled Table
$Enc((KA_1, KB_0), KO_0)$
$Enc((KA_0, KB_1), KO_0)$
$Enc((KA_1, KB_1), KO_1)$
$Enc((KA_0, KB_0), KO_0)$

Garbled Circuits - Example

- Millionaire's Problem
- Alice and Bob want to find out who's richer
- But they don't want to reveal their wealth to each other
- Can use garbled circuit to compute ">" on their private inputs

Garbled Circuits - Millionaire's Problem

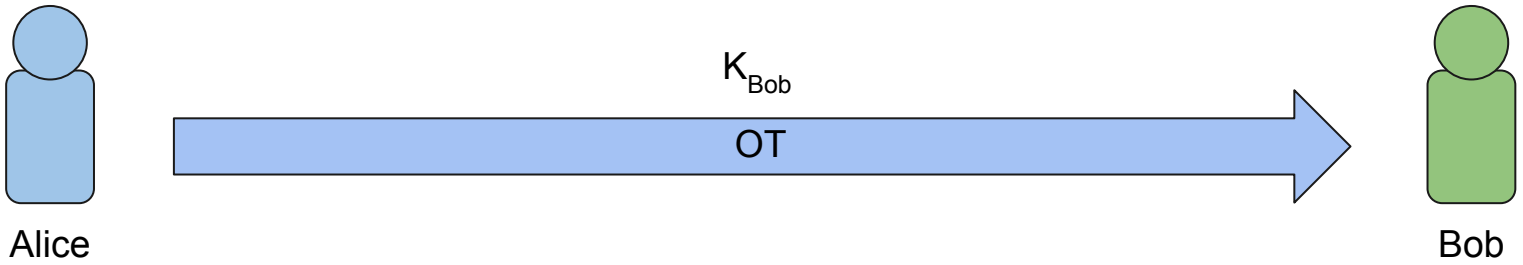
Generates garbled circuit C for ">" function
Sends Bob keys for her inputs K_{Alice}



However, Bob still needs the keys for his inputs!

Garbled Circuits - Millionaire's Problem

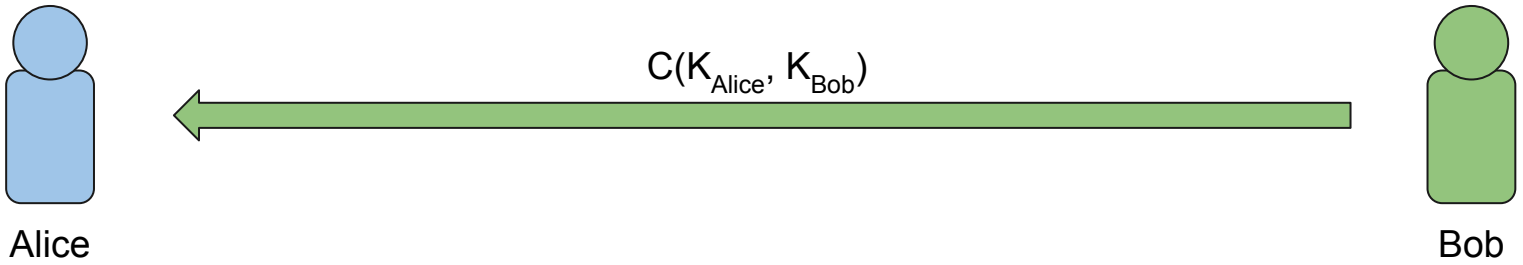
Bob requests keys for his inputs K_{Bob} using OT



- Alice does not learn Bob's input
- Bob does not learn Alice's input

Garbled Circuits - Millionaire's Problem

Evaluates garbled circuit
Shares result with Alice



Zero Knowledge Proofs (ZKP)

- How much knowledge needs to be transmitted to prove a statement T?
- ZKP: a way to prove that a statement is true without revealing any other information
    - Only need 1 Bit!

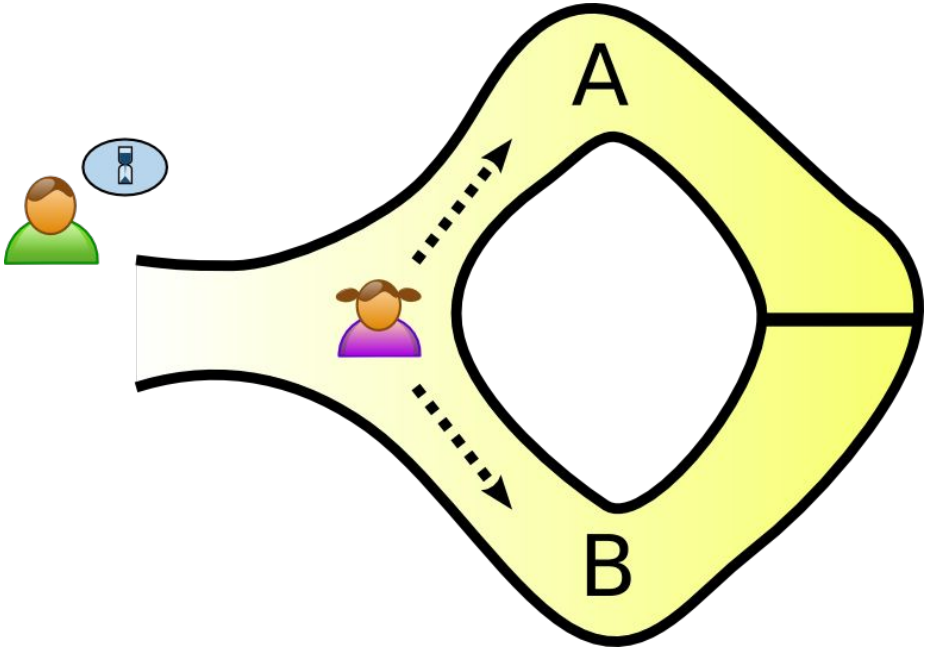
Zero Knowledge Proofs (ZKP) - Properties

- Completeness - an interactive proof protocol is complete if given an honest prover and honest verifier, the protocol succeeds with overwhelming probability. (Verifier accepts prover's claim)
- Soundness - If the Prover is cheating, they cannot convince an honest verifier that the claim is true except with negligible probability
- Zero Knowledge - If the verifier is cheating - they do not learn anything other than the statement is true (if the statement is true).
- Not deterministic, Probabilistic!
- Interactive!

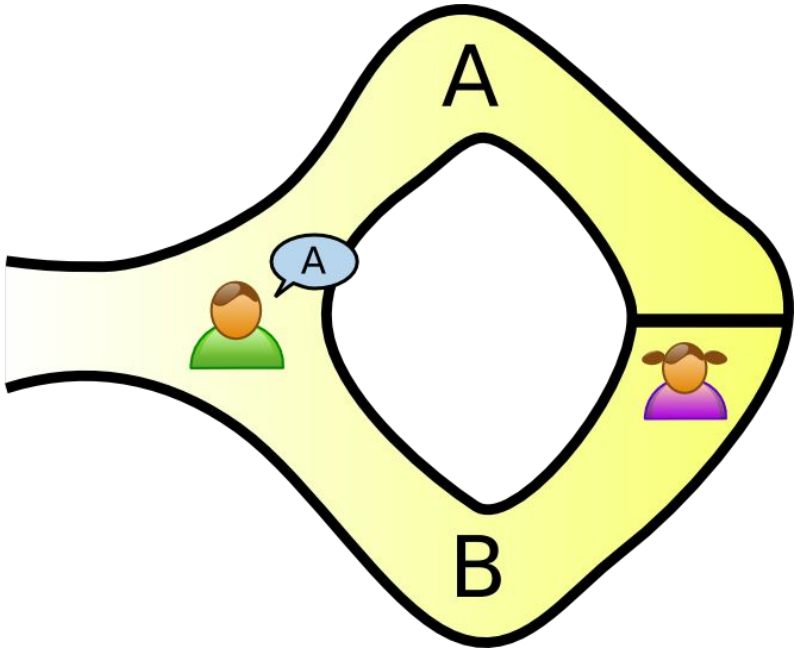
ZKP - Story Time!

- The Cave of Ali-Baba
- Peggy the Prover, knows the secret password to a door in a circular cave
- Viktor the Verifier wants to know the secret word, willing to pay
- How can I show you that I know this secret word without:
  - You inherently trusting that I do
  - Opening the door in front of you
- Zero Knowledge Proofs to the Rescue!

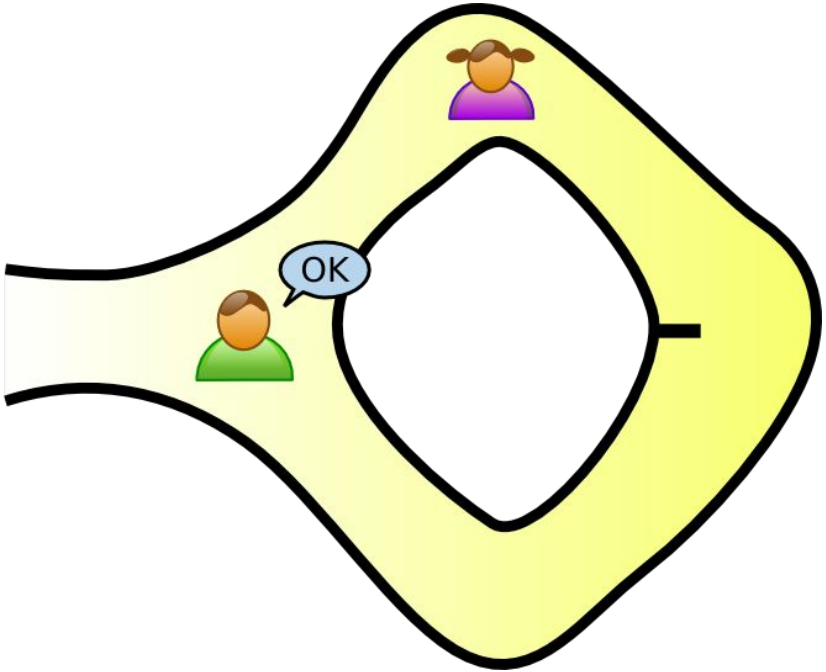
ZKP - Story-Time!



ZKP - Story-Time!



ZKP - Story-Time!



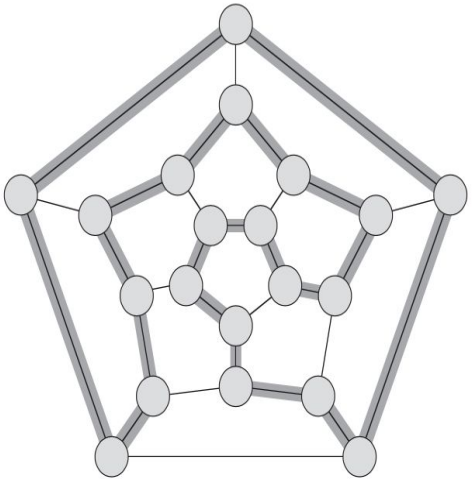
ZKP - Ali Baba Lessons

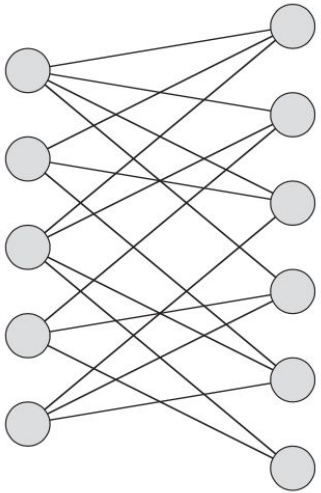
- What happens if Peggy Doesn't actually know the secret word?
  - Cheating with 50% chance of success
- Let's run the protocol n times: Peggy now has a 2^-n chance of cheating without being caught
- Peggy has to 'commit' to a value so she can't cheat (walk down one of the cave paths)
- No external party learns anything from this game
  - We could have cheated, together!

ZKP - More Realistically

- How can we translate this simple example into something more concrete?
- **Hamiltonian Cycles:**
  - Hamiltonian Path: path that traverses each vertex exactly one time
  - Hamiltonian Cycle: Hamiltonian Path that also forms a cycle
  - NP Complete Problem!

Hamiltonian Cycles





ZKP - Hamiltonian Cycle (Manuel Blum)

- Peggy wants to prove that she knows a Hamiltonian Cycle in a large graph G without revealing the cycle itself
- Viktor wants to verify that Peggy knows the cycle, but Peggy has no intentions of letting her secret out

ZKP - Hamiltonian Cycle: Protocol

- Peggy takes G and builds H, which is isomorphic to G
  - Peggy knows the isomorphism of the graph, so she trivially can compute the hamiltonian cycle
- Peggy “Commits” to H (Puts it face down on the table)
- Viktor now has two choices
  - Ask for the isomorphism between G and H
  - Ask for the hamiltonian cycle in H

ZKP - Hamiltonian Cycle: Lessons

- Why is this specific example important?
  - Hamiltonian Cycle problem is NP-Complete
- This ZKP implies that there are zero knowledge proof systems for all NP-Complete Problems!

ZKP - Let's Not Interact

- Clearly the 'interactive' portion of the proof requires two parties to be present
- We can avoid this!
  - Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (ZK-Snarks)
  - "Succinct" - milliseconds to verify, at most hundreds of bytes to prove
  - Single message from prover to verifier
- Currently only way to produce proofs of these type is to have initial setup phase that generates a reference string common to the prover and verifier
- Uses oblivious polynomial evaluation and partially homomorphic encryption as part of the verification process.

Private Set Intersection (PSI)

- Find the intersection of a set of N values without revealing any information about values outside the intersection
- Relies on oblivious function evaluation:
  - Multiple parties jointly compute a function.
  - Their inputs are private and cannot be derived from the function computation
  - Dishonest parties should not be able to get any additional information about other parties' inputs
  - Typically very computationally expensive

Private Set Intersection (PSI)

Story Time: Two graduate students, Charles and Tong, are working on a take-home test.

- As honest students, they would never give answers to each other.
- But they would like to double check their answers with each other to make sure they do well.

So they come up with a loop-hole:

It's not cheating if the other student already knows an answer. So they perform PSI with their answers -- if the intersection is not the same as their sets, they know that they disagree on a question.

Private Set Intersection (PSI)

Some Constraints:

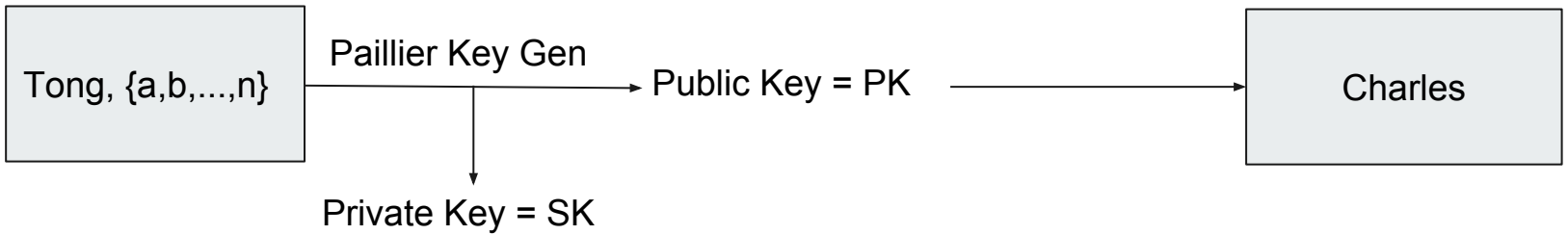
- These are sets -- there are no duplicates!
- Only positive numbers. If there are negatives, we have to transform them to positives.

Building Blocks:

- Partial Homomorphic Encryption (PHE)
- Oblivious Polynomial Evaluation (OPE)

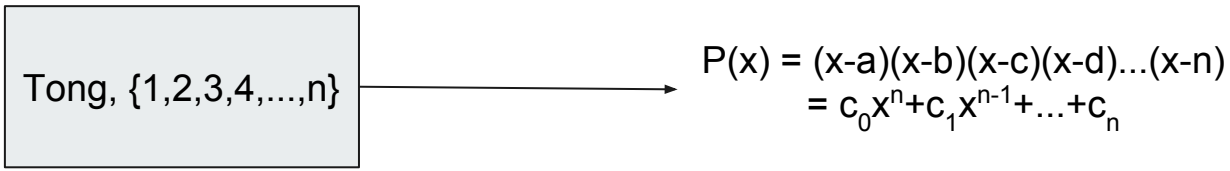
Private Set Intersection (PSI)

1. Tong generates a PHE compatible public/private key pair (Paillier Encryption Scheme) and sends the public key to Charles.
    - A homomorphic function is one such that f(a+b) = f(a)+f(b)



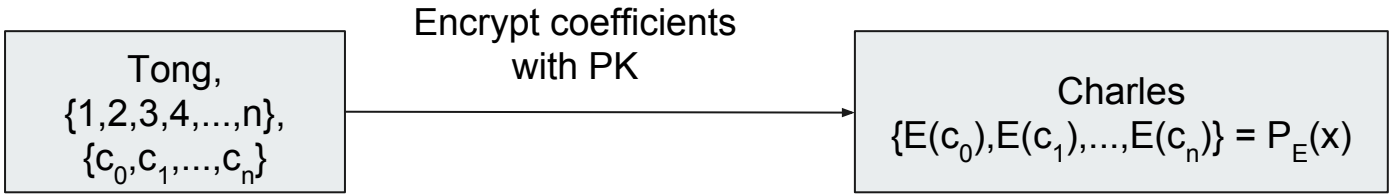
Private Set Intersection (PSI)

2. Tong generates a polynomial P(x) with roots at his set elements. The coefficients are sufficient for polynomial evaluation.



Private Set Intersection (PSI)

3. Tong encrypts the coefficients of P(x) with his **public key** and sends them to Charles.
   - **Charles cannot decrypt the coefficients without knowledge of the private key**
   - **Let the polynomial defined by the encrypted coefficients be $P_E(x)$**

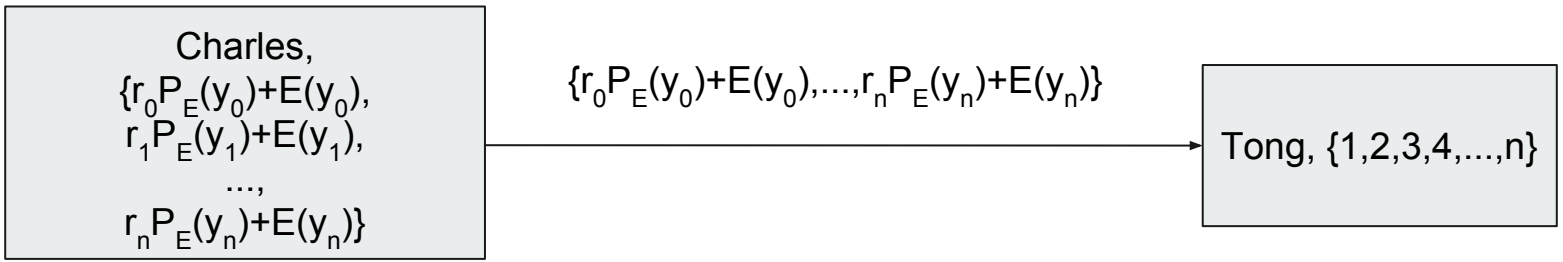


Private Set Intersection (PSI)

4. Charles encrypts each y in his set and computes $r^*P_E(y)+E(y)$, for random r.
    - Since the encryption is homomorphic, Charles is actually computing $E(r^*P(y)+y)$

Private Set Intersection (PSI)

5. Charles sends the E(r*P(y)+y) to Tong, who decrypts them. If any of the decrypted numbers are in Tong's set, then they are also in Charles's set.



Private Set Intersection (PSI)

Why does this work?

- Since the encryption is additively homomorphic $r*P_E(y)+E(y)$ is equivalent to $E(r*P(y)+y)$
- If any element y in Charles's set is also in Tong's set, then y is a root of P(x)
- Evaluation of P(x) at a root is 0!

Therefore, for the common elements z in their sets, $E(r*P(z)+z) = E(0+z) = E(z)$. So, when Tong decrypts, he learns that z is also in Charles's set.

Charles cannot learn the other elements of Tong's set, though, because the roots have become encrypted

Topics

- Overview
- History
- Algorithms
- **Applications**
- Implementations
- Open Issues
- Discussion

MPC Applications

- Computation on sensitive or private data
  - Healthcare or finance analytics
- Cryptographic voting schemes - private votes, public election results
- Predicting satellite collisions

ZKP Applications

- Interactive ZKP
  - Fiat-Shamir identification protocol
- NonInteractive ZKP
  - Zcash
  - Ethereum

Private Set Intersection Applications

- Private Consensus/Voting
  - Several parties have candidates for something (maybe an organ donor recipient, job contract, etc).
  - Use PSI to see if the parties agree on the recipient
- Secure Information Sync
- Secure Mutual Friend List -- [Insert Facebook Joke Here]
- And of course, not-cheating on take-home tests...

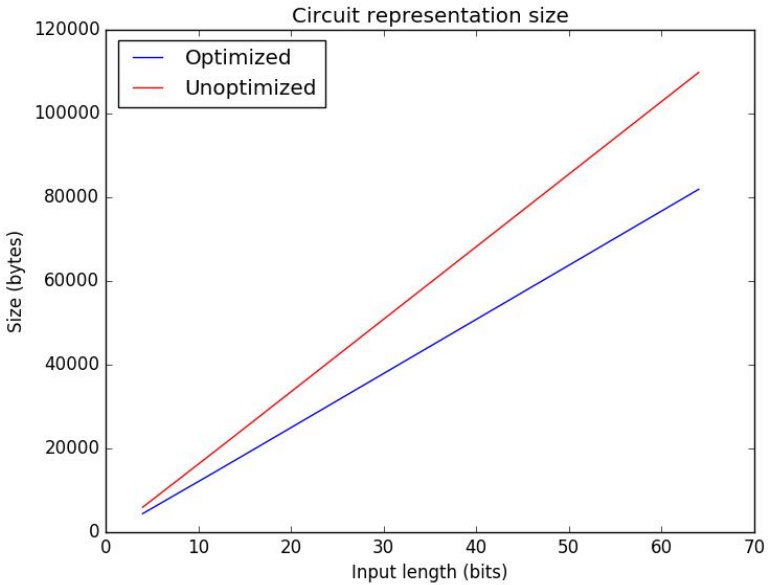
Topics

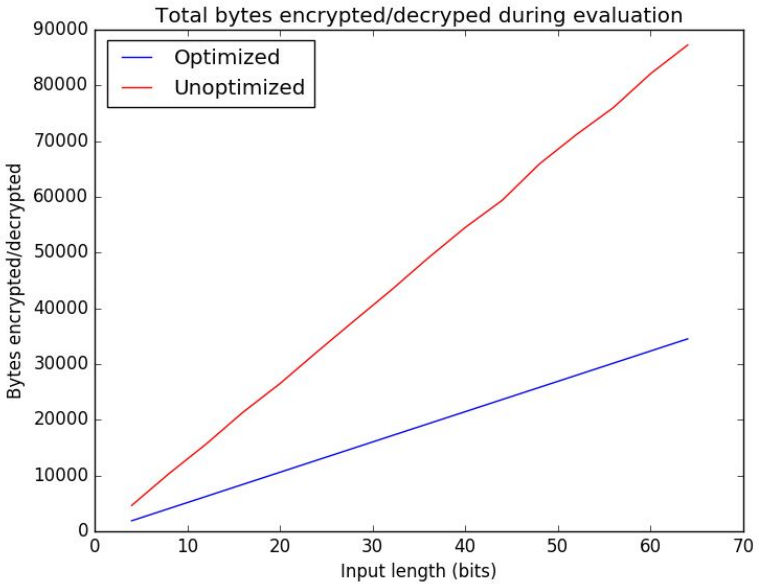
- Overview
- History
- Algorithms
- Applications
- **Implementations**
- Open Issues
- Discussion

Implementations - Garbled Circuits

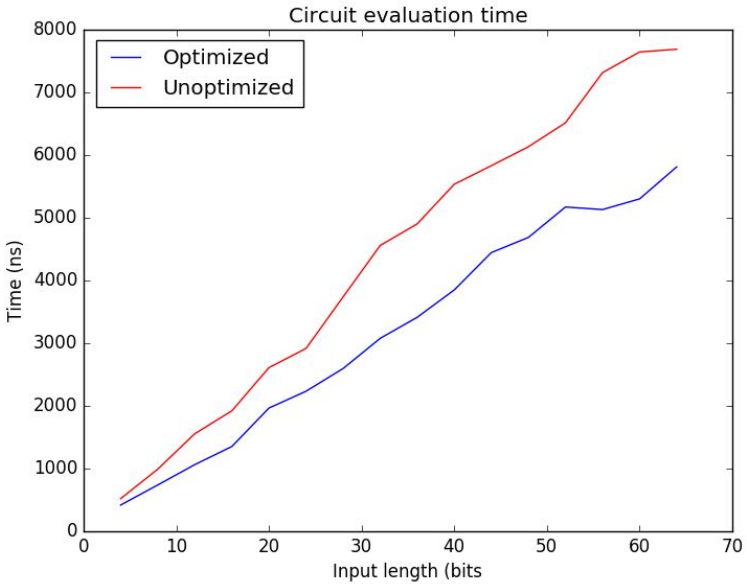
- Implemented garbled circuits with oblivious transfers
- Evaluated effect of point-and-permute optimization
- Tested simple comparison circuit (Millionaire's problem)
- Key measures:
  - Evaluation time
  - Circuit generation time
  - Circuit representation size
  - Bytes encrypted/decrypted

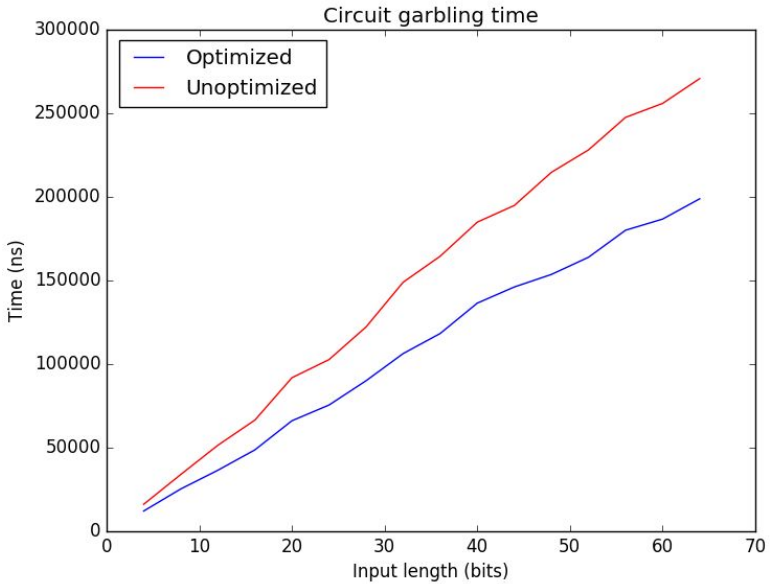
Implementations - Garbled Circuits





Implementations - Garbled Circuits

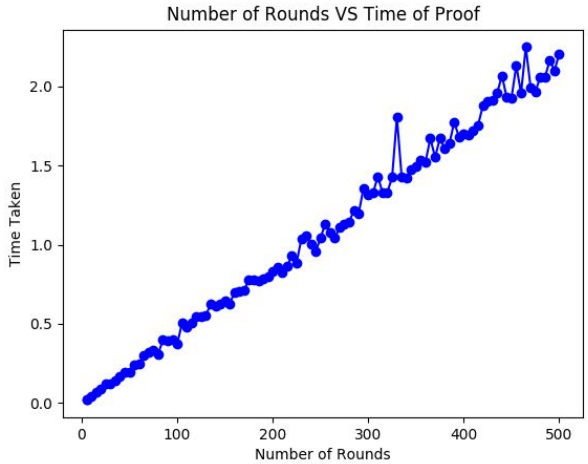


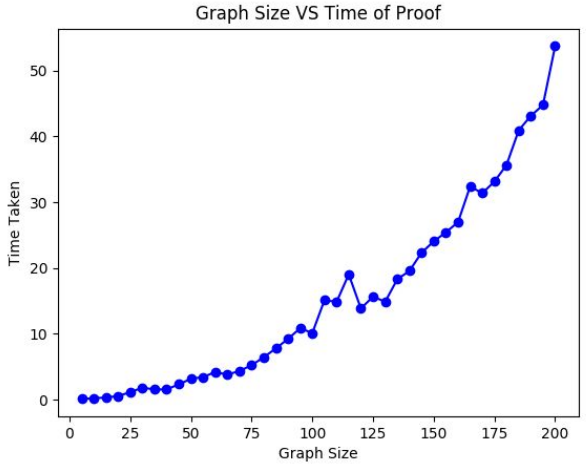


Implementations - ZKP

- Implemented the hamiltonian Cycle ZKP in Python
  - Includes cryptographic commitment scheme for messages by Peggy!
- In our Hamiltonian Cycle ZKP Simulator we've come up and implemented the following commitment scheme:
  - 1. Cryptographically Hash Possible Messages
  - 2. Encrypt Hashes and Send over to Viktor
  - 3. Viktor Asks question
  - 4. Peggy Encrypts and sends answer to question
  - 5. Viktor Decrypts and checks that the hash of the decryption matches the corresponding original hash

ZKP - Boring Graphs - Or Not?





Implementations - PSI

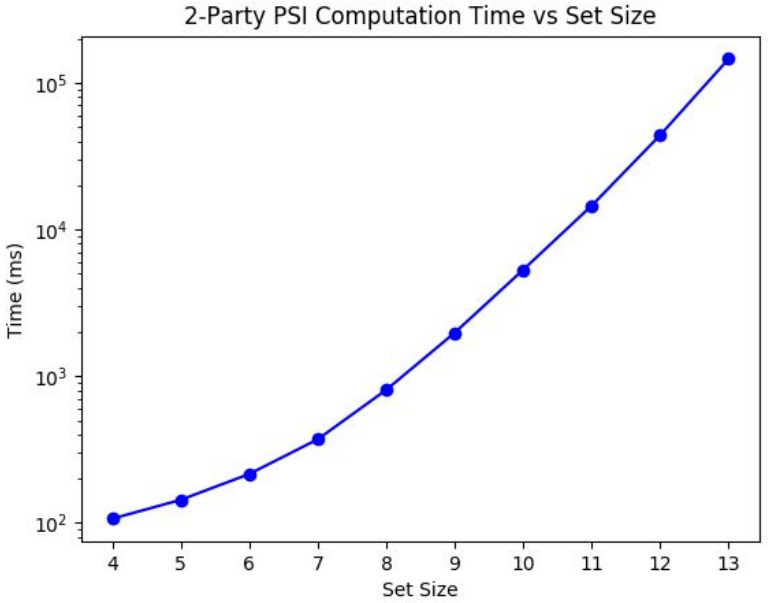
- Initial implementation only permitted 2 parties:
  - Relies on **Partial Homomorphic Encryption** and **Oblivious Polynomial Evaluation**
- Newest PSI schemes allow for any number of parties:
  - Relies on **Oblivious Transfer** and **Oblivious Programmable Pseudorandom Functions**
- **These are still extremely computationally intensive!**

Private Set Intersection (PSI)

To the right is the run time of PSI using a naive implementation in Python using Phaillier PHE -- **Note the log scale**

Yikes! That's still not linear even with the log scale!

Can we beat the curve by using Horner's rule for OPE?



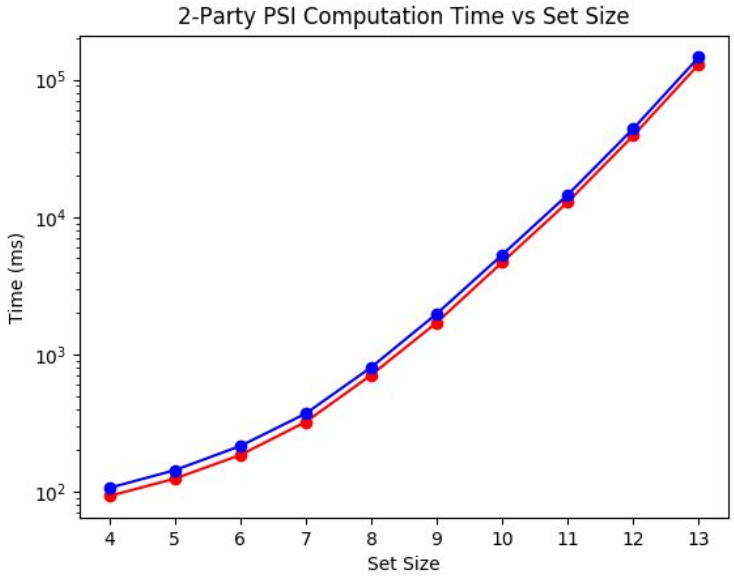
Private Set Intersection (PSI)

The red line shows times while using Horner's Rule for the OPE.

Horner's Rule hardly made a dent! What gives?

The numbers are large so that PHE operations on them are expensive.

Most papers report them in terms of evaluations of exponentials



Private Set Intersection

- Fortunately, there are some better (still not scalable) alternatives and primitives:
  - Symmetric Key Variants -- Avoid Modular Exponentiation
  - Oblivious Programmable Pseudorandom Functions -- Avoids Polynomial Interpolation and Evaluation
  - Other Primitives: Oblivious Bloom Filters with Cuckoo Hashing, etc.
- Still not scalable enough for large scale operation, though.

Private Set Intersection (PSI)

Protocol	Communication		Computation		Corruption	Security
	Leader	Client	Leader	Client	Threshold	Model
[24]	$\mathcal{O}(tnm\log(\lvert X\rvert))\lambda$		$\mathcal{O}(ntm^2)$		$n-1$	semi-honest
[6]	$\mathcal{O}((n^2m+nm)\lambda)$		$\mathcal{O}(nm+m)$		$\lfloor n/2 \rfloor$	semi-honest
[17]	$\mathcal{O}(nm\lambda)$	$\mathcal{O}(m\lambda)$	$\mathcal{O}(mn\log_2(m))$	$\mathcal{O}(m)$	$n-1$	semi-honest
Ours	$\mathcal{O}(nm\lambda)$	$\mathcal{O}(m\lambda)$	$\mathcal{O}(n\kappa)$	$\mathcal{O}(\kappa)$	$n-1$	augmented semi-honest
		$\mathcal{O}(mt\lambda)$		$\mathcal{O}(t\kappa)$		semi-honest

Table 1: Communication (bits) and computation (number of exponentiations) complexities of multi-party PSI protocols in the semi-honest setting, where n is number of parties, t dishonestly colluding, each with set size m; X is the domain of the element; and λ and κ are the statistical and computational security parameters, respectively. In our protocols, the computational complexities are in an offline preprocessing phase.

Kolesnikov, Vladimir, et al. "Practical multi-party private set intersection from symmetric-key techniques."

Topics

- Overview
- History
- Algorithms
- Applications
- Implementations
- **Open Issues**
- Discussion

Open Issues

- Efficient fully-homomorphic encryption (FHE)
  - Allows *arbitrary* computation on encrypted data
  - Does not require Alice to provide a circuit for the function
  - Some schemes exist, but computations are very inefficient
- Scalable Computations
  - MPC relies on functions that can produce correct results from transformed user inputs
  - The transformed user inputs are typically large so that they are hard to reverse
  - Evaluating the function takes more time due to the size of the inputs
  - Sometimes additional steps must be taken to ensure correctness

Topics

- Overview
- History
- Algorithms
- Applications
- Implementations
- Open Issues
- **Discussion**

Key References

- Andrew Yao. Protocols for Secure Computations (Extended Abstract). In FOCS ’82, pages 160–164, 1982.
- Michael O. Rabin. "How to exchange secrets by oblivious transfer." Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.
- S Goldwasser, S Micali, and C Rackoff. 1985. The knowledge complexity of interactive proof-systems. In Proceedings of the seventeenth annual ACM symposium on Theory of computing (STOC '85). ACM, New York, NY, USA, 291-304. DOI=http://dx.doi.org/10.1145/22145.22178
- Freedman, Michael J., Kobbi Nissim, and Benny Pinkas. "Efficient private matching and set intersection." International conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 2004.

Discussion

-

Test Questions

1. How much data does a zero knowledge proof emit?
2. If a function f is additively homomorphic, what is f(a)+f(b) in terms of f?
3. In oblivious transfer, what does the sender learn? The receiver?

Thank you!

Additional References

- Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In STOC, pages 103–112, 1988.
- Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. Communications of the ACM, 28(6):637–647, 1985.
- Yao, Andrew Chi-Chih (1986). "How to generate and exchange secrets". Foundations of Computer Science, 1986., 27th Annual Symposium on. IEEE: 162–167.
- Quisquater JJ. et al. (1990) How to Explain Zero-Knowledge Protocols to Your Children. In: Brassard G. (eds) Advances in Cryptology — CRYPTO' 89 Proceedings. CRYPTO 1989. Lecture Notes in Computer Science, vol 435. Springer, New York, NY
- https://z.cash/technology/zksnarks.html
- Adi Shamir, R. L. Rivest, and Leonard M. Adleman. Mental poker. Technical Report LCS/TR-125, Massachusetts Institute of Technology, April 1979.

Additional References (Cont.)

- Manuel Blum. Three applications of the oblivious transfer: Part I: Coin flipping by telephone; part II: How to exchange secrets; part III: How to send certified electronic mail. Technical report, University of California, Berkeley, CA, USA, 1981.
- Goldreich, Oded; Micali, Silvio; Wigderson, Avi (1987). "How to play ANY mental game". Proceeding STOC '87 Proceedings of the nineteenth annual ACM symposium on Theory of computing. ACM: 218–229.
- Songhori, Ebrahim M; Hussain, Siam U; Sadeghi, Ahmad-Reza; Schneider, Thomas; Koushanfar, Farinaz (2015). "TinyGarble: Highly compressed and scalable sequential garbled circuits". Security and Privacy (SP), 2015 IEEE Symposium on. IEEE: 411–428.
- Kolesnikov, Vladimir, et al. "Practical multi-party private set intersection from symmetric-key techniques." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.

Additional References (Cont.)

- Feige, U., Fiat, A. & Shamir, A. J. Cryptology (1988) 1: 77. https://doi.org/10.1007/BF02351717
- Blum, Manuel. "How to prove a theorem so no one else can claim it." *Proceedings of the International Congress of Mathematicians*. Vol. 1. 1986.

Image References

- "File:Zkip alibaba1.png." *Wikimedia Commons, the free media repository*. 27 Jan 2018, 18:47 UTC. 29 Mar 2018, 02:40 <https://commons.wikimedia.org/w/index.php?title=File:Zkip_alibaba1.png&oldid=283253751>.
- "File:Zkip alibaba2.png." *Wikimedia Commons, the free media repository*. 27 Jan 2018, 18:47 UTC. 29 Mar 2018, 02:40 <https://commons.wikimedia.org/w/index.php?title=File:Zkip_alibaba1.png&oldid=283253751>.
- "File:Zkip alibaba3.png." *Wikimedia Commons, the free media repository*. 27 Jan 2018, 18:47 UTC. 29 Mar 2018, 02:40 <https://commons.wikimedia.org/w/index.php?title=File:Zkip_alibaba1.png&oldid=283253751>.