

# The IPS Compiler: Optimizations, Variants and Concrete Efficiency

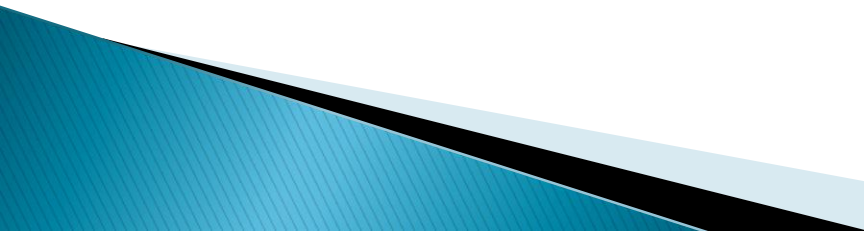
Yehuda Lindell, Benny Pinkas and Eli Oxman  
Bar-Ilan University, Israel

# Secure Computation Settings

## ▶ Information theoretic

- Uses aesthetic mathematical tools that are typically very efficient
- Adversary is computationally unbounded
- Requires honest majority

## ▶ Computational

- Uses computational hardness for oblivious transfer, zero knowledge and more
  - Adversary runs in polynomial time
  - Any number of corrupted parties
- 

# Adversary Models

- ▶ **Semi-honest**

- Corrupted parties follow protocol, but try to learn more than allowed by inspecting transcript

- ▶ **Malicious**

- Corrupted parties follow any arbitrary strategy

- ▶ **Covert**

- Corrupted parties follow any strategy
- If they follow a strategy enabling them to cheat, then they are guaranteed to be caught with some probability (e.g.,  $\frac{1}{2}$ )

# A Construction Paradigm [GMW]

- ▶ **Step 1** – construct a protocol that is secure for **semi-honest** adversaries
- ▶ **Step 2** – construct a **compiler** that transforms any protocol that is secure for semi-honest adversaries into a protocol that is secure for **malicious** adversaries
- ▶ The GMW87 compiler achieves step 2 by using zero-knowledge proofs (and more) to ensure semi-honest behaviour

# The IPS Compiler

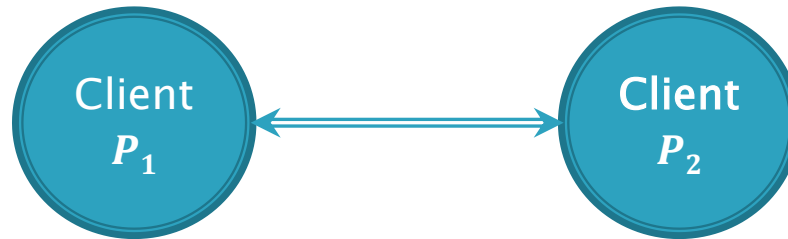
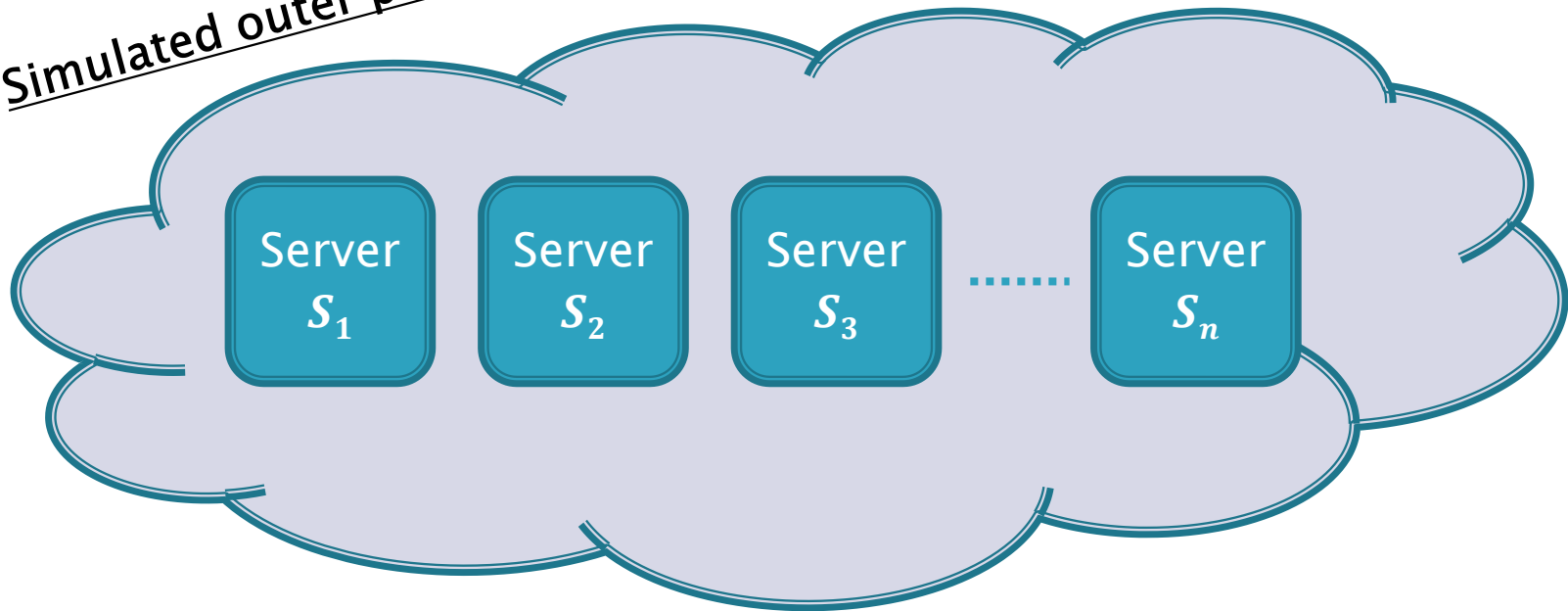
- ▶ At Crypto 2008, Ishai et al. presented a completely different compiler for obtaining security for any number of corrupted parties
- ▶ The building blocks of IPS
  - An information-theoretically secure protocol for computing the functionality (secure for **malicious**)
  - **Semi-honest** protocols for computing simple functions (like shares of the product of shares)
- ▶ **Advantages of IPS**
  - Excellent asymptotic efficiency
  - Completely different way of working
  - Black-box in the semi-honest protocols

# The Basic Idea

- ▶ Simulate an information-theoretic protocol that is secure for an honest majority (malicious adversary)
  - Let  $\Pi$  be an information-theoretic protocol for  $n$  parties/servers ( $n$  is a parameter to be determined)
- ▶ A real multiparty protocol for  $m$  parties (with  $m < n$ ) works by having the  $m$  real parties **simulate** an execution of  $\Pi$ 
  - The  $m$  parties run secure protocols  $\pi_1, \dots, \pi_n$  where  $\pi_i$  is a secure simulation of the  $i^{\text{th}}$  server
- ▶ Servers are virtual and  $\Pi$  is called the outer protocol
- ▶ The  $m$  real parties are called clients and  $\pi_1, \dots, \pi_n$  are called inner protocols

# Server-Simulation by Clients

Simulated outer protocol



Real inner protocols  $\pi_1, \dots, \pi_n$ ; Server  $S_i$  is simulated with inner protocol  $\pi_i$

# Server Simulation by Clients

- ▶ **What security level is required by the inner protocols  $\pi_1, \dots, \pi_n$ ?**
  - If they are secure against malicious, this is clearly fine
  - However, our aim is to use subprotocols that are secure for weaker (say, semi-honest) adversaries
  - If they are secure for only semi-honest, then what stops a real malicious client from cheating?



# Security of Inner Protocols

- ▶ Consider inner protocols  $\pi_1, \dots, \pi_n$  that are secure for **covert** adversaries
  - With any cheating detected with probability  $\frac{1}{2}$
- ▶ In order to cheat in the outer protocol  $\Pi$  (which is secure as long as only a minority are corrupt), the adversary has to cheat in at least  $n/2$  inner protocols
  - Cheating in an inner protocol is the only way to “corrupt” a server in the outer simulated protocol  $\Pi$
- ▶ By the covert guarantees, such cheating will go undetected with probability at most  $2^{-n/2}$
- ▶ The protocol is therefore **secure for malicious adversaries**

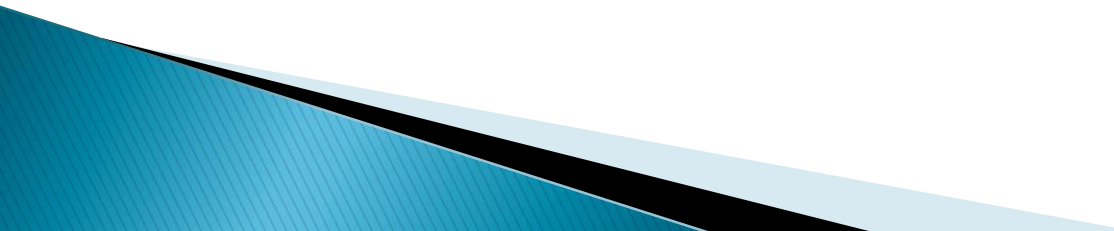
# Starting from Semi-Honest

- ▶ **The challenge:** how to prevent a malicious party from cheating in a semi-honest protocol
- ▶ **Watching:** if the randomness (and inputs) that should be used by one party is known to the others, then any cheating can be detected
- ▶ **The IPS watchlist mechanism:**
  - Each party “watches” every other party in  $k$  out of the  $n$  (real) inner protocols
  - No party knows where it’s being watched (oblivious transfer based setup)
  - Therefore, cheating in many inner protocols is detected with high probability (like covert)

# Our Results

- ▶ **We study the IPS compiler from a number of different angles**
  - **Optimizations:** we provide efficiency improvements on the IPS construction
  - **Variants:** we apply the IPS paradigm to study covert security and its relation to both semi-honest and malicious adversaries
  - **Concrete efficiency:** we calculate the concrete efficiency of IPS (in contrast to just asymptotic)

# Optimizations

- ▶ **More efficient watchlist setup protocol**
    - Based on DDH; uses a special committed oblivious transfer type of protocol
    - Our protocol also gives a more exact result, enabling a tighter cheating probability (yielding better concrete efficiency)
    - Our setup is much more efficient and allows for the use of more servers (which can be in the **thousands**)
  
  - ▶ **More in the paper...**
- 

# Variants

- ▶ **IPS constructs malicious from semi-honest**
- ▶ **We use the IPS paradigm to:**
  - Construct covert from semi-honest
    - Just like IPS but with few watchlists
  - Construct malicious from covert
    - As we saw before
- ▶ **Significance**
  - Deepen understanding of covert adversary model (open question from TCC 2010)
  - Conceptually and technically simple
  - Better asymptotic efficiency for some problems

# Concrete Efficiency

- ▶ **IPS has been shown to have excellent asymptotic efficiency, but no one knows how it behaves concretely**
  - This is due to the high level of abstraction
  - Efficiency depends on:
    - The outer information-theoretic protocol used
    - The inner protocols used
    - The number of servers and watchlists to obtain a given error

# Choosing the Outer Protocol

- ▶ **All multiplication gates require an interactive inner protocol**
  - Best efficiency is therefore achieved by minimizing the number of multiplications
  - This is achieved using the packed secret sharing methodology
- ▶ **Note that the most efficient information-theoretic protocol is not necessarily optimal here**

# Number of Servers – Analysis

- ▶ **The smallest number of servers possible should give the best efficiency**
  - Less work in simulating the outer protocol
- ▶ **However, less servers means less corruptions needed by the adversary to achieve an effective dishonest majority**
  - And so more watchlists to catch cheating
  - And in turn more servers to maintain an honest majority
- ▶ **Instantiating IPS concretely and efficiently requires choose these parameters optimally**



# Number of Servers – Analysis

- ▶ We carry out an analytic and numerical analysis of optimal parameters for IPS for a number of different circuits
- ▶ We have some rather surprising results
  - For example, for the case of 2 parties and an outer protocol secure for a plain honest majority  $4k$  servers is optimal ( $3k$  results in effectively more servers for the same error probability)
    - Recall  $k$  is the number of watchlists

# Difficulties

- ▶ One of the major difficulties with the IPS protocol is that its instantiation is different
  - For every **function** (circuit)
    - The circuit size and structure affects the choice of block size (for packed secret sharing), affecting the degree of the polynomial, affecting the number of servers and the size of the watchlists and so on
      - The number of servers can in turn affect the circuit, unless the circuit is over a huge field to start with
  - For every **number of clients**
- ▶ Analyzing the optimal number of servers, watchlist size and so on is a very difficult task

# A Concrete Count

- ▶ **AES-type circuit (2400 gates over 100 layers)**
  - A minimal number of OT's and multiplications is achieved by taking block size  $n/73$  (numerical analysis)
- ▶ **For this block size (and protocol threshold) we found “optimal” parameters for error  $2^{-40}$ :**
  - Number of servers  $n=1752$
  - Number of watchlists  $k=207$
- ▶ **The actual cost (for 2 different choices of the inner multiplication protocol)**
  - 13.8 million OT's and 4.5 billion field multiplications
  - 5.5 million OT's and 5.5 billion field multiplications
- ▶ **What's better? It probably depends on the machine...**

# Experimental Results

## ▶ Caveats:

- The estimates are based on only a partial implementation (see full paper for details)
- The AES circuit is over  $\text{GF}[2^8]$  but we have many more than 256 servers, so actually need secret sharing over a field extension (which hasn't been studied concretely)

## ▶ Time estimates

- Using software-based field multiplications the time estimate is about 950 seconds
- Using the new Intel AES chip which gives carry-less multiplications, the time estimate is reduced to between 79 and 94 seconds (probably a bit **overly optimistic**)

## ▶ Surprisingly competitive (and no real attempts to fully optimize the protocol)

# Conclusions – IPS Efficiency

- ▶ **From our concrete analysis, we believe that IPS may actually be concretely competitive**
  - Even more potential for multiparty where efficient alternatives are less common
- ▶ **There are serious obstacles and difficulties in implementing IPS**
  - There is no general protocol that receives a circuit and works (the parameters must be tailored)
  - But the payoff may be worth it, and more research may yield a way of doing this...

# Summary

- ▶ **A deeper understanding of the IPS compiler**
  - IPS and covert adversaries
  - Optimized watchlist setup
    - More efficient but also cleaner security analysis
  - Better parameters for IPS
- ▶ **IPS and efficiency/practicality**
  - Very difficult to specify and implement, but may potentially yield competitive protocols
  - More work is needed for understanding concrete costs and for optimizing for specific protocols
  - New optimizations may further improve situation
    - Like our new watchlist setup