# Secure Multi-Party Computation of Probabilistic Threat Propagation

**Emily Shen**
**Nabil Schear, Ellen Vitercik, Arkady Yerukhimovich**

**Graph Exploitation Symposium 2016**

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY


This material is based upon work supported by the Assistant Secretary of Defense for Research and Engineering under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Assistant Secretary of Defense for Research and Engineering.

# Problem: Collaborative Cyber Defense

**Companies should share cyber threat information with each other and USG to prevent attacks**

**Sector-Specific Sharing Centers**

FINANCIAL SERVICES Information Sharing and Analysis Center

DIB ISAC
DEFENSE INDUSTRIAL BASE
INFORMATION SHARING AND ANALYSIS CENTER

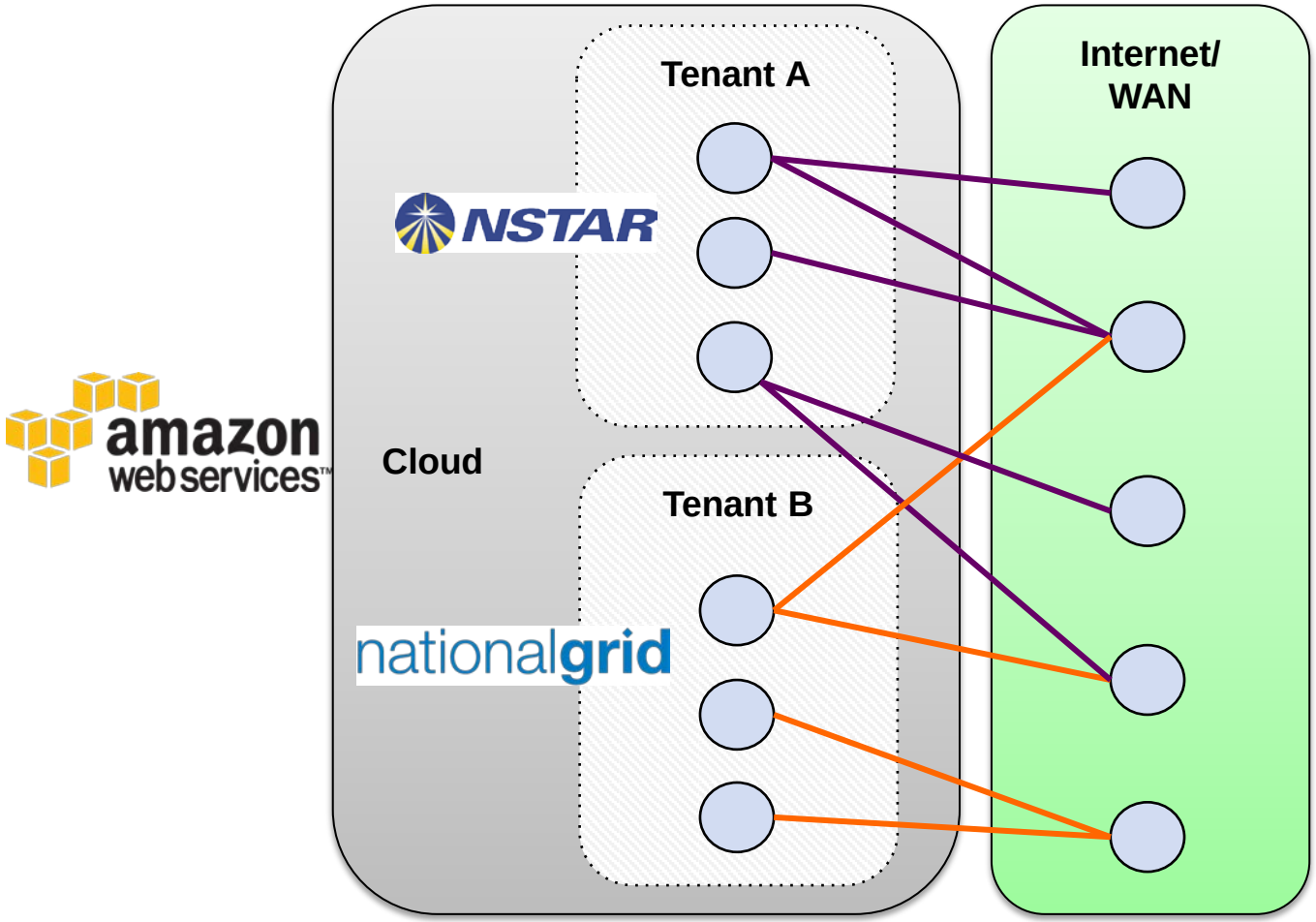
AVIATION ISAC

HEALTH ISAC

- **Global cyber situational awareness benefits all**
- **However, cybersecurity information is sensitive**
  - **Prior attacks may indicate weaknesses in defense**
  - **Business information could be used by competitors**

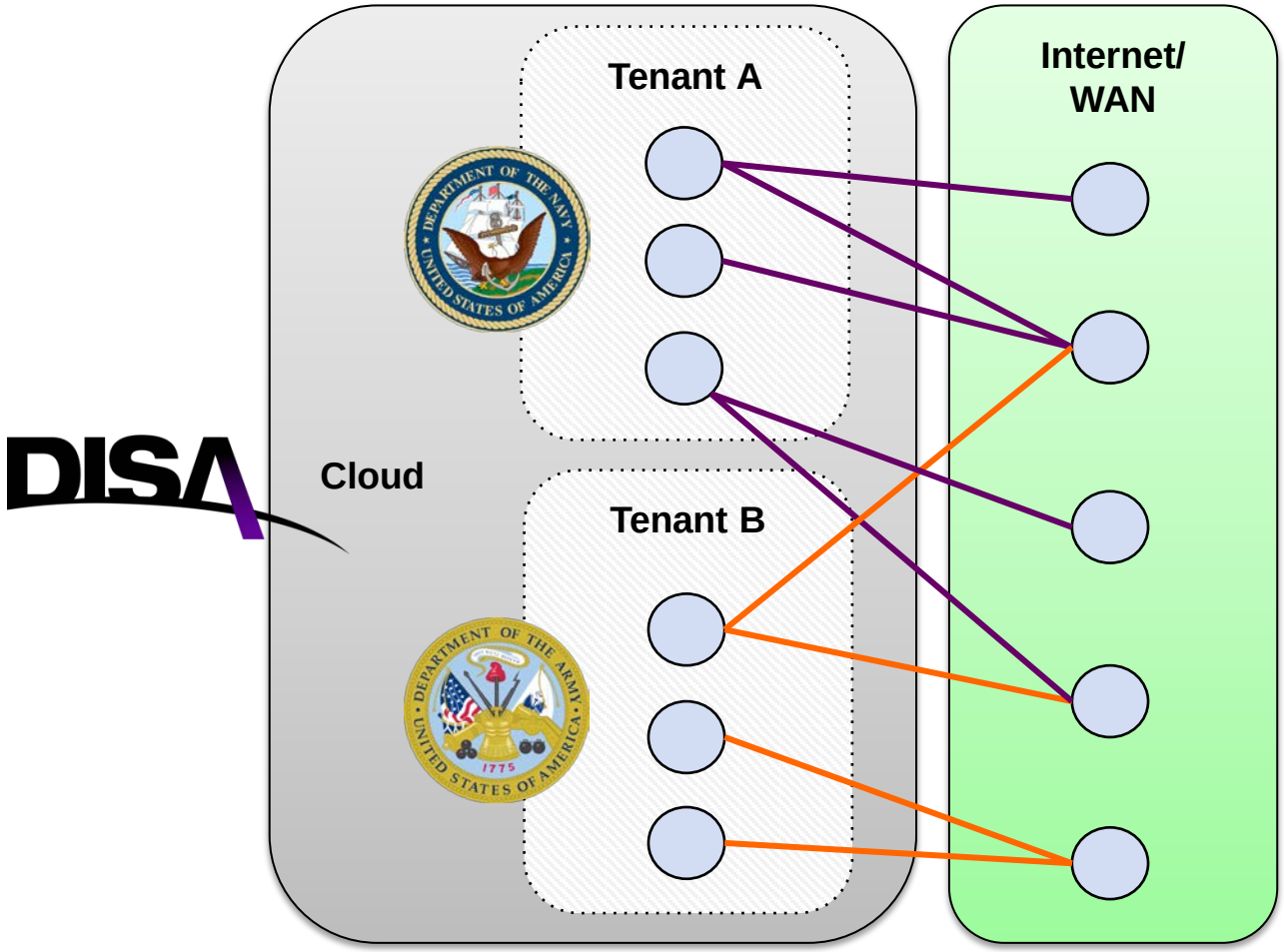
**Goal: Compute joint analytics without sharing sensitive cyber security information**

# Cloud Threat Sharing



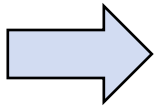

**Cloud introduces provider and multiple tenants with varying trust**

# Cloud Threat Sharing



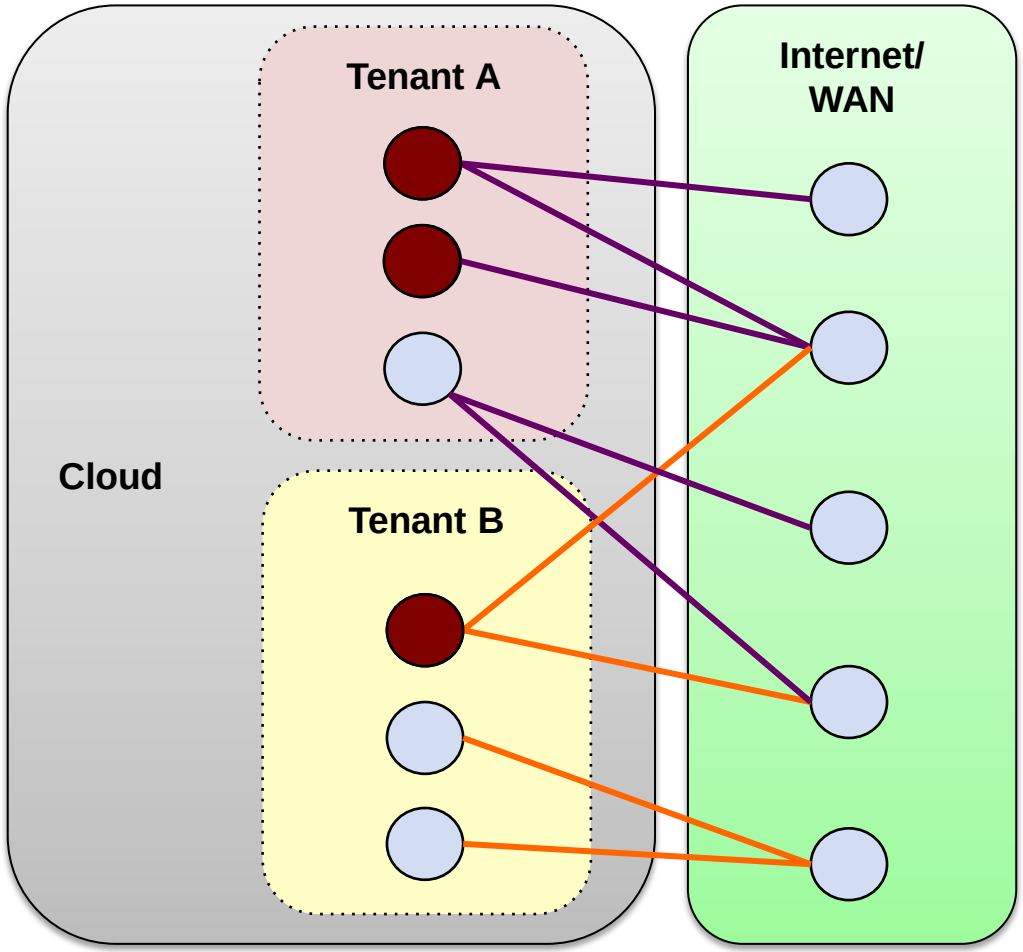

**Cloud introduces provider and multiple tenants with varying trust**

# Outline



- **Motivation**
- **Probabilistic Threat Propagation (PTP)**
- **Secure Multi-Party Computation (MPC)**
- **Application of MPC to PTP**

# Probabilistic Threat Propagation (PTP)



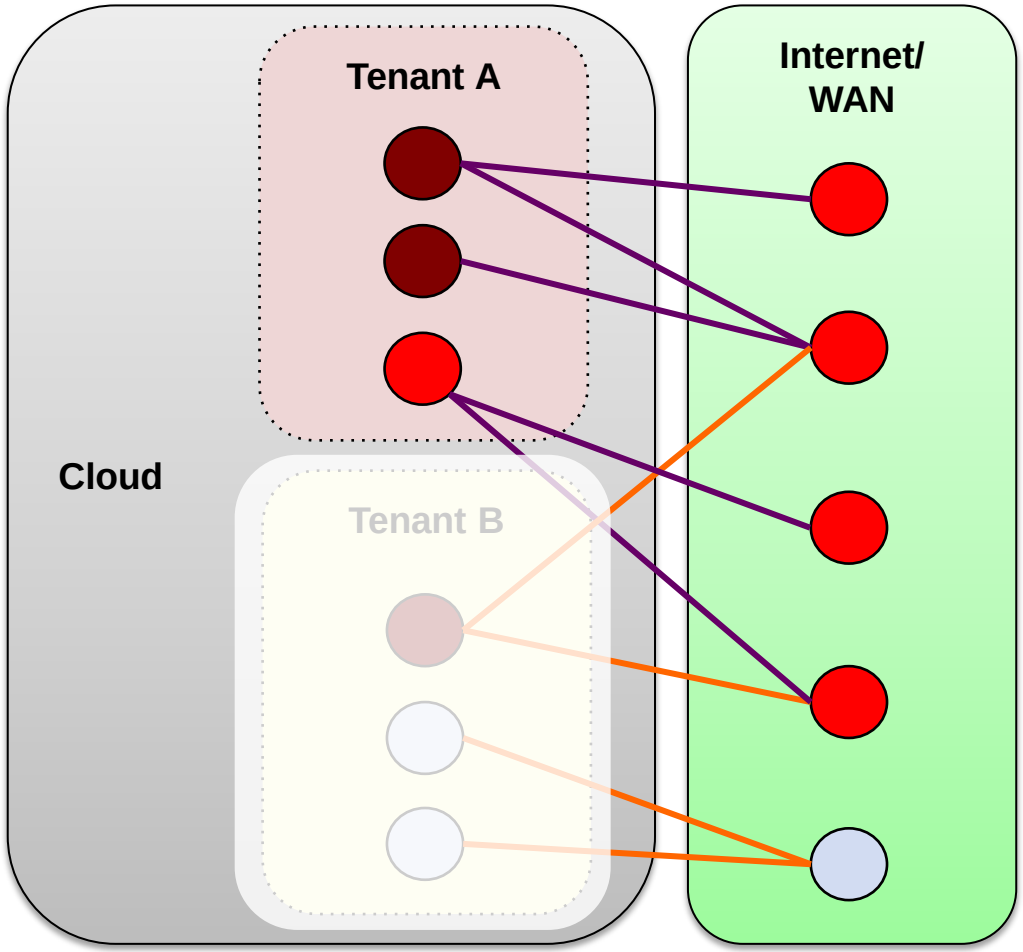

**Require:** $W$, $\{tips\}$, $\gamma$, $\alpha$
$P \leftarrow \alpha^N$, $P(\{tips\}) \leftarrow \gamma$
$C \leftarrow 0^{N \times N}$
**repeat**
$T \leftarrow W \otimes P^T$
$C \leftarrow T - W \circ C^T$
$P \leftarrow < C, \bar{1} >$
$C(\{tips\}, \cdot) \leftarrow 0$
$P(\{tips\}) \leftarrow \gamma$
**until** $P$ has converged
**return** $P$

**Goal: Use connectivity graph and known bad hosts to predict other compromised hosts**



Kevin M. Carter, Nwokedi C. Idika, William W. Streilein: Probabilistic Threat Propagation for Network Security. IEEE Transactions on Information Forensics and Security 9(9): 1394-1405 (2014)

# Probabilistic Threat Propagation (PTP)



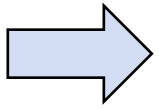

**Require:** $W$, $\{tips\}$, $\gamma$, $\alpha$
$P \leftarrow \alpha^N$, $P(\{tips\}) \leftarrow \gamma$
$C \leftarrow 0^{N \times N}$
**repeat**
$T \leftarrow W \otimes P^T$
$C \leftarrow T - W \circ C^T$
$P \leftarrow < C, \bar{1} >$
$C(\{tips\}, \cdot) \leftarrow 0$
$P(\{tips\}) \leftarrow \gamma$
**until** $P$ has converged
**return** $P$

**Tenants don't want to share compromised host info with each other or with provider**



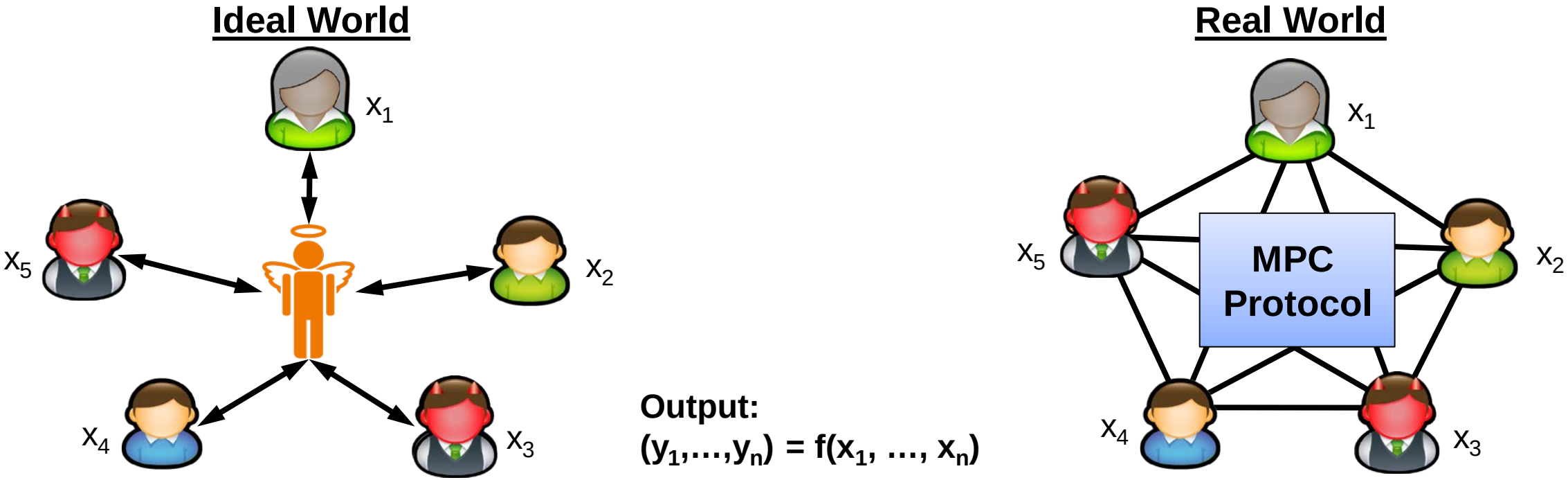
Kevin M. Carter, Nwokedi C. Idika, William W. Streilein: Probabilistic Threat Propagation for Network Security. IEEE Transactions on Information Forensics and Security 9(9): 1394-1405 (2014)

# Outline



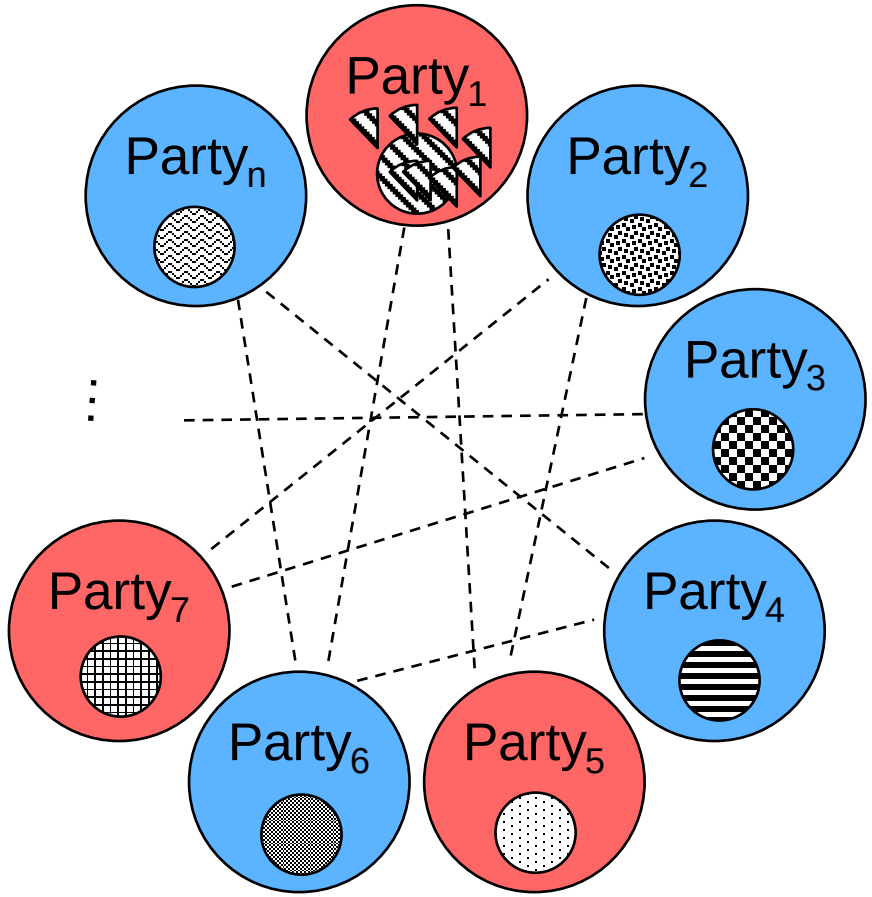
- **Motivation**
- **Probabilistic Threat Propagation (PTP)**
- **Secure Multi-Party Computation (MPC)**
- **Application of MPC to PTP**

# Secure Multi-Party Computation (MPC)




- **MPC emulates a trusted party, assuming no more than a threshold number of adversarial parties, guaranteeing:**
  - **Correctness of computation**
  - **Confidentiality of inputs and outputs**

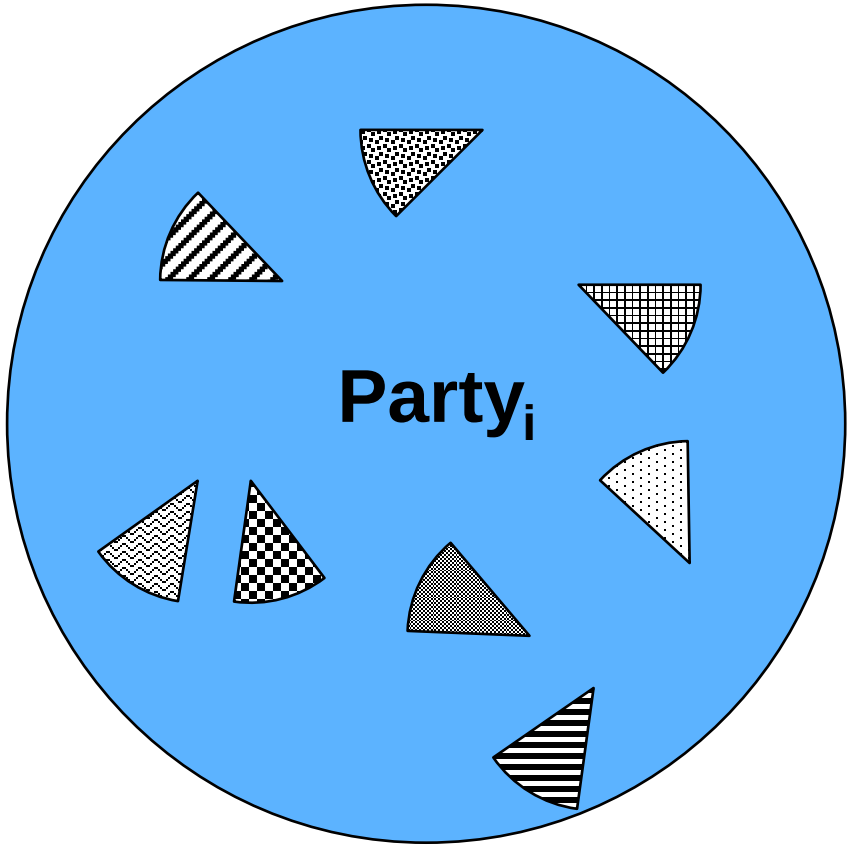
# Secret Sharing Overview




- **Each secret value can be “split” into pieces (“shares”)**
- **If enough people combine shares, they can reconstruct the secret**
- **Otherwise, no one learns any info**

Slide credit: Sasha Berkoff

# MPC Based on Secret Sharing




**To perform MPC, each party:**

- **Sends shares of its secret to each other party**
- **Computes on shares**
  - **Addition: local computation**
  - **Multiplication: requires interaction**
- **Interacts to combine manipulated shares to obtain final answer**

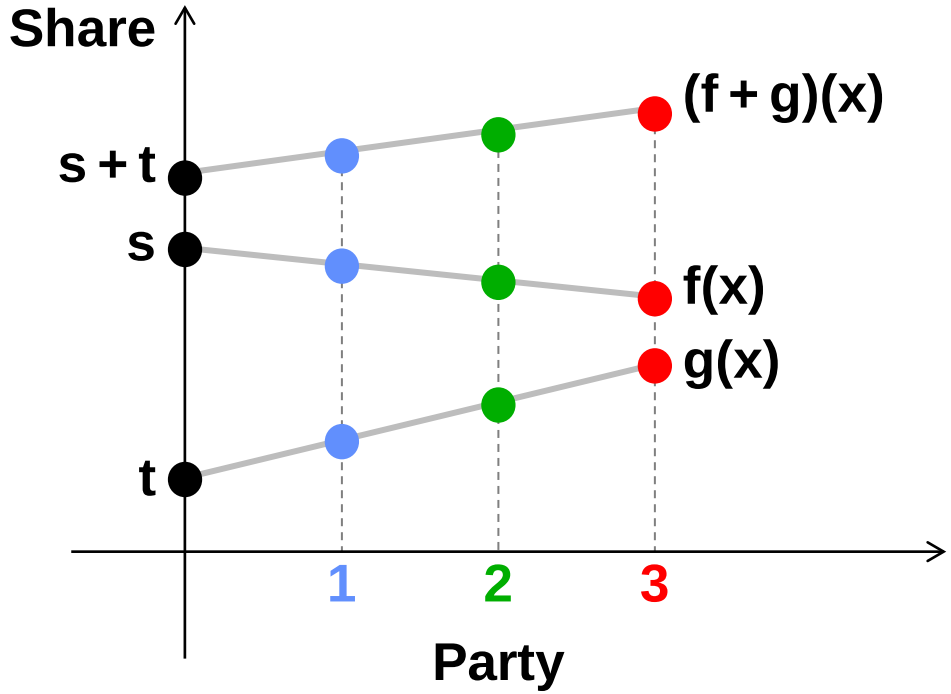
Slide credit: Sasha Berkoff

# Secret Sharing Example

**To secret share for a threshold k:**

- **Give each party a point on a random degree-k polynomial**
- **Reconstruction: interpolate any >k points to recover degree-k polynomial**
- **Addition (s + t): f(i) + g(i)**
- **Multiplication (s × t): f(i) × g(i), then interactive degree reduction**

**Example: threshold k = 1**

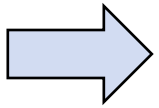



**Secret sharing can be used to compute any arithmetic function securely**



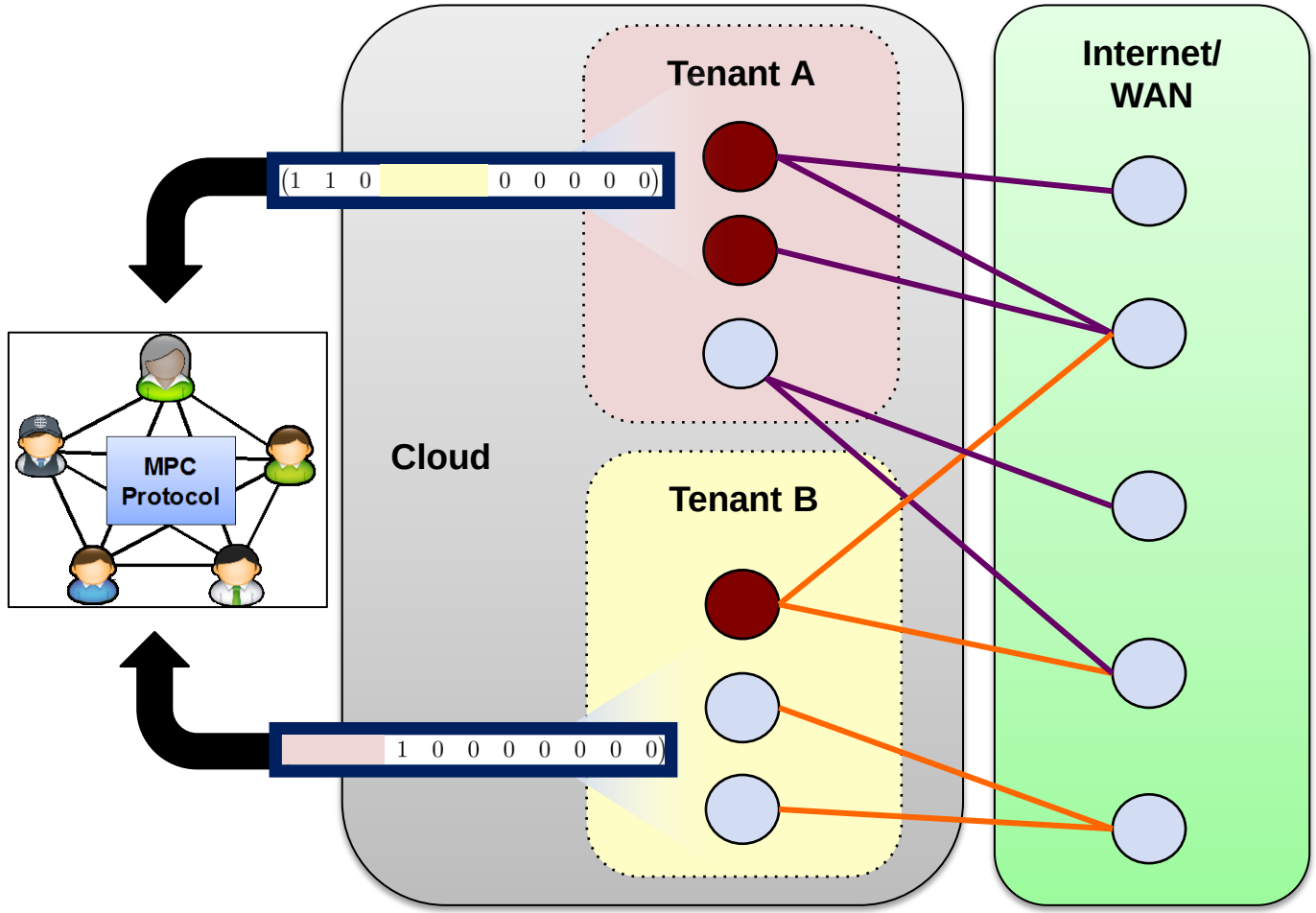
BGW – Ben-or, Goldwasser, Wigderson: "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract)." STOC 1988

# Outline



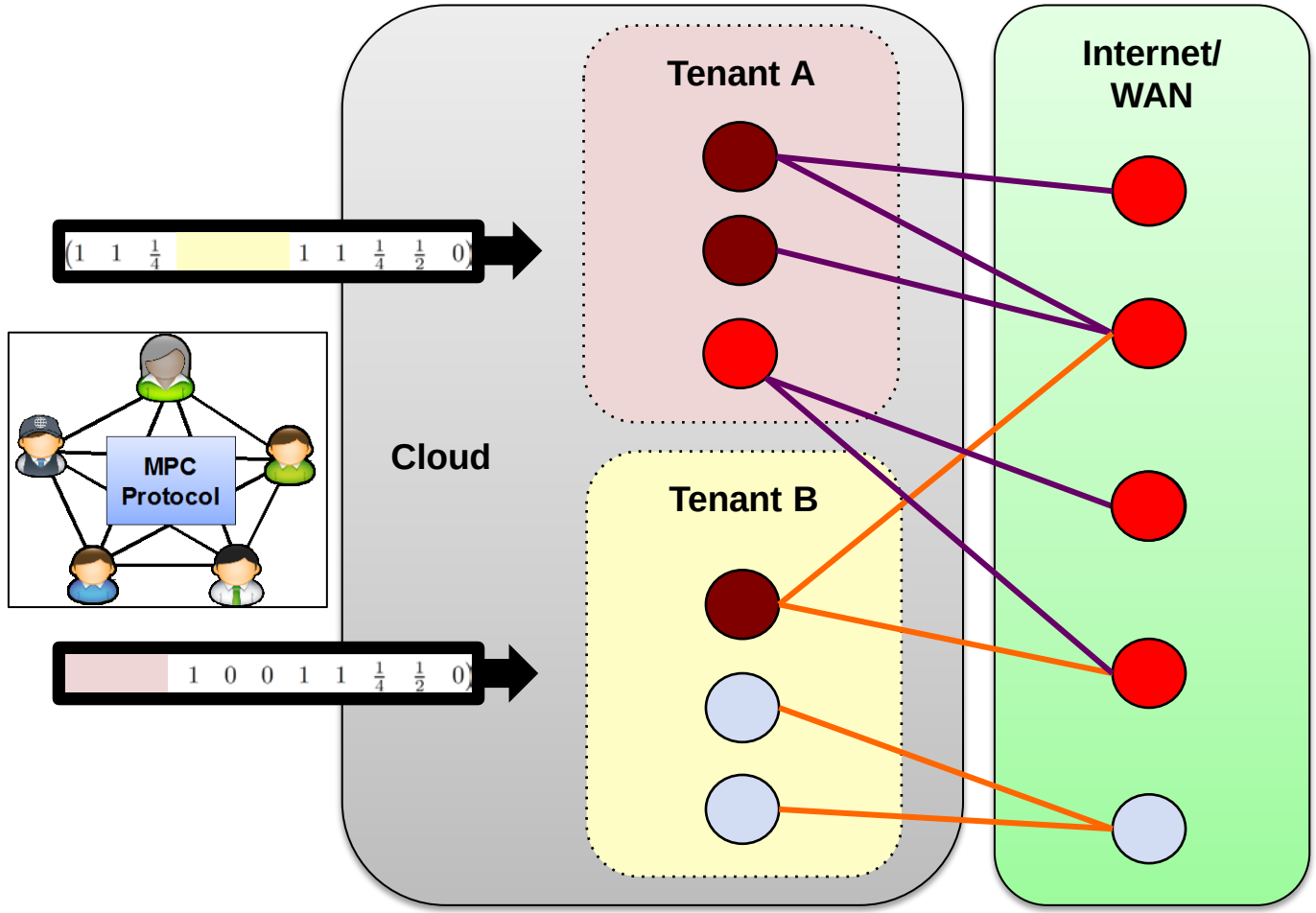
- **Motivation**
- **Probabilistic Threat Propagation (PTP)**
- **Secure Multi-Party Computation (MPC)**
- **Application of MPC to PTP**

# Secure Multi-Party Computation of PTP




**Tenants secret-share compromised host info with other tenants, jointly compute PTP securely**

# Secure Multi-Party Computation of PTP




**Each party learns about compromised hosts in its own network and on the Internet**

# Secure Multi-Party Computation of PTP

## Probabilistic Threat Propagation Algorithm

**Require:** $W$, $\{tips\}$, $\gamma$, $\alpha$
$P \leftarrow \alpha^N$, $P(\{tips\}) \leftarrow \gamma$
$C \leftarrow 0^{N \times N}$
**repeat**
$T \leftarrow W \otimes P^T$
$C \leftarrow T - W \circ C^T$
$P \leftarrow < C, \bar{1} >$
$C(\{tips\}, \cdot) \leftarrow 0$
$P(\{tips\}) \leftarrow \gamma$
**until** $P$ has converged
**return** $P$

MPC Operations

- $N^2$ mult-by-const
- $N^2$ mult-by-const, adds
- $N^2$ adds
- $N^2$ mults
- N mults, N adds
- N mults, 2N adds, 1 comp

- **Map high-level algorithm to mathematical operations**
- **Low overhead: add; multiply by constants**
- **High overhead: multiply, compare**

# MPC Optimization Techniques

## 1 Fixed Point Arithmetic

**Efficiently compute on non-integer values using secure fixed point arithmetic**

| Representation | $a \to \tilde{a} := \lfloor a \cdot 2^{30} \rfloor$ |
|---|---|
| Addition | $a+b \to \tilde{a}+\tilde{b} \approx \lfloor (a+b)\cdot 2^{30} \rfloor$ |
| Multiplication | $a \cdot b \to \tilde{a}\cdot\tilde{b}/2^{30} \approx \lfloor (a\cdot 2^{30})\cdot(b\cdot 2^{30})/2^{30} \rfloor = \lfloor (a\cdot b)\cdot 2^{30} \rfloor$ |

**All MPC computation done on integer values**

## 2 Oblivious Selection

**Select/assign values to/from secret indices using secret-shared indicator vector**

$$\begin{pmatrix} 1/3 \\ 0 \\ 2/3 \end{pmatrix} * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2/3 \end{pmatrix}$$

$$\underset{\text{Original shares}}{\begin{pmatrix} 92134 \\ 41431 \\ 19384 \end{pmatrix}} * \underset{\text{Indicator vector}}{\begin{pmatrix} 14321 \\ 3413 \\ 7324 \end{pmatrix}} = \underset{\text{New shares}}{\begin{pmatrix} 34549 \\ 3412 \\ 23412 \end{pmatrix}}$$

**MPC hides selection/assignment indices**

## 3 Sparse Matrices

**Reduce computation on sparse matrices by storing only nonzero entries**

**Original**

$$\begin{pmatrix} 0 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 6 \\ 2 & 0 & 0 & 2 & 2 \\ 3 & 0 & 3 & 0 & 0 \\ 0 & 3 & 3 & 0 & 0 \end{pmatrix}$$

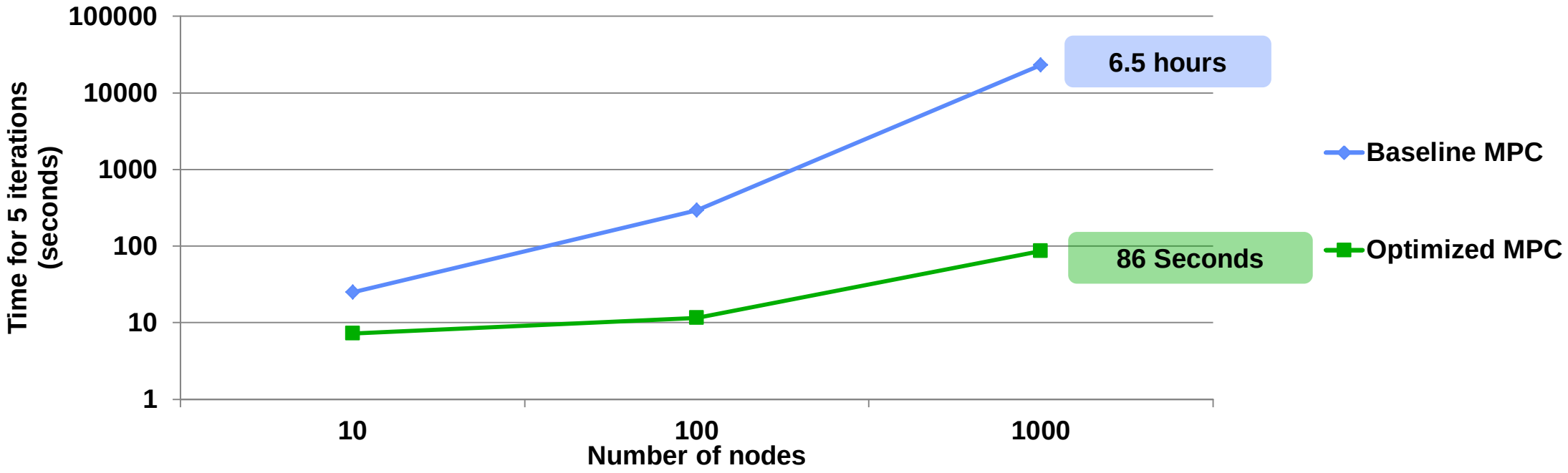
**Sparse**

$$\begin{pmatrix} 3 & 3 & 6 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 2 & 3 & 3 & 3 & 4 & 4 & 5 & 5 \\ 3 & 4 & 5 & 1 & 4 & 5 & 1 & 3 & 2 & 3 \end{pmatrix}$$

**Nonzeros**
**Row indices**
**Col indices**

**MPC operations only needed on nonzero entries**

# Performance of Secure Multi-Party Computation of PTP




**Optimizations result in >250× speedup over baseline MPC**

# Summary

- Graph analytics (e.g., probabilistic threat propagation) have important applications to cyber security
- Privacy concerns may restrict ability to perform joint analytics
- Secure multi-party computation enables privacy-preserving computation of analytics
- Designed and optimized MPC for PTP, achieving significant speedup over baseline MPC
- Future work: design and optimize secure computation other useful graph analytics