Almost-everywhere Secure Computation

Juan Garay  (Bell Labs)
Rafail Ostrovsky  (UCLA)
Secure Multi-party Computation (MPC)

**Multi-party computation** (MPC) [Goldreich-Micali-Wigderson 87] :
- \( n \) parties \( \{P_1, P_2, \ldots, P_n\} \), \( t \) corrupted; each \( P_i \) holds a private input \( x_i \)
- One public function \( f(x_1, x_2, \ldots, x_n) \)
- All want to learn \( y = f(x_1, x_2, \ldots, x_n) \) (Correctness)
- Nobody wants to disclose his private input (Privacy)

**2-party computation** (2PC) [Yao 82] : \( n=2 \)
MPC: Network Requirements

Unconditional (information-theoretic) MPC [BGW88, CCD88]:

n players, t corrupted, n > 3t
MPC: Network Requirements

Unconditional (information-theoretic) MPC [BGW88, CCD88]:
- $n$ players, $t$ corrupted, $n > 3t$
MPC on Incomplete Networks
MPC on Incomplete Networks

Small (constant) connectivity!

“doomed”

“privileged”
MPC on Incomplete Networks: Butterfly
MPC on Incomplete Networks: Butterfly
MPC on Incomplete Networks: Butterfly
MPC on Incomplete Networks: Butterfly
This Work: Almost-everywhere MPC

- “Give up” some of the players; guarantee security for a large fraction of them
- Adv. implicitly wiretaps by corrupting sufficiently many neighbors
- Capture privacy requirement
  - Definitional effort
  - Adaptive adversaries
- $G_n = (V,E)$
This Work: *Almost-everywhere MPC*

- “Give up” some of the players; guarantee security for a large fraction of them
- Adv. *implicitly* wiretaps by corrupting sufficiently many neighbors
- Capture **privacy** requirement
  - Definitional effort
  - *Adaptive* adversaries
- $G_n = (V,E)$
This Work: *Almost-everywhere* MPC

- “Give up” some of the players; guarantee security for a large fraction of them
- Adv. *implicitly* wiretaps by corrupting sufficiently many neighbors
- Capture **privacy** requirement
  - Definitional effort
  - *Adaptive* adversaries
- \( G_n = (V,E) \)
- \( W: \textit{Privileged} \) set

\[
\begin{align*}
V \quad &\quad X(T) \\
W \quad &\quad T
\end{align*}
\]
Almost-everywhere MPC (cont’d)

- \( G = (V,E), \ |T| = t, \ P = 2^V \)

- \( \chi : P^{(\leq t)} \rightarrow P \)
  1. \( T_1 \subseteq T_2 \Rightarrow \chi(T_1) \subseteq \chi(T_2) \)
  2. \( T \subseteq \chi(T) \)
  
  \( \chi = \max_T \{\chi(T)\} \)

- Protocol \( \Pi \) achieves \textbf{X-MPC} if \( \exists W, |W| \geq n - X \), s.t. all players in \( W \) are able to perform MPC

- Fully connected network: \( \chi(T) = T \)
“Commit-and-Compute” Paradigm

A two-phase protocol $\Pi$ achieves X-MPC if for any PPT function $F$ the following are satisfied

1. **Commit phase:** Players in $V$ commit to their inputs
   - **Binding:** For all $P_i \in V$ there is uniquely defined $x^*_i$
   - **Privacy:** For all $P_i \in W$, $x^*_i$ is information-theoretically hidden

2. **Computation phase:**
   - **Correctness:** For all $P_i \in W$, $P_i$ outputs $F(x^*_1, x^*_2, \ldots, x^*_n)$
   - **Privacy:** For all $X^*_W, Y^*_W, Z^*_X(T)$ such that
     \[ F(X^*_W, Z^*_X(T)) = F(Y^*_W, Z^*_X(T)) \]
   - the adversary can’t distinguish $\Pi(X^*_W, Z^*_X(T))$ from $\Pi(Y^*_W, Z^*_X(T))$
X-MPC Protocols: Preview

General strategy:

- Large privileged set $W$, $X = n - |W|$
- Endow players in $W$ with resources needed (in fully connected networks) for unconditional MPC
- Require $X < n/3 \Rightarrow$ MPC on $W$
Talk Plan

- Secure multi-party computation (MPC)
- *Almost-everywhere* MPC (X-MPC)
- Related work
- Tools & ingredients
- X-MPC protocols
Talk Plan

- Secure multi-party computation (MPC)
- *Almost-everywhere* MPC (X-MPC)
- Related work
- Tools & ingredients
- X-MPC protocols
Related Work

- “Almost-everywhere agreement” [DPPU86, BG90, Upf92]
- Perfectly secure message transmission (PSMT) [DDWY89,…]
  - $(2t+1)$-connectivity for reliable and private comm.
- Privacy amplification/secret key agreement [BBR88,BBCM95,…]
  - Authentic public channel + private (corrupted) channel
- “Hybrid” corruptions [GP92, FHM98]
  - Adv. actively corrupts some players, wiretaps others
- Secure computation on incomplete networks [Vaya07]
Talk Plan

- Secure multi-party computation (MPC)
- *Almost-everywhere* MPC (X-MPC)
- Related work
- Tools & ingredients
- X-MPC protocols
X-MPC: Ingredients

- “Almost-everywhere agreement” [DPPU86]
  Players in W can implement a broadcast channel
  $\rightarrow$ Almost-everywhere $(i,t)$-admissible graphs

- Secure message transmission (SMT) [DDWY89]
  by public discussion
  $\rightarrow$ Obtain pair-wise secure channels between nodes in W

- Verifiable secret sharing (VSS) [CGMA85]
  $\rightarrow$ Implement Commit phase of X-MPC
Broadcast (aka Byz. agreement) [PSL80, LSP82]

- n players, t corrupted
- Value v
- Source
- $v_1$, $v_2$, $v_3$, ..., $v_{n-1}$

- If source is honest, $v_i = v$ (Validity)
- $v_i = v_j$ (Agreement)

$n > 3t$
Almost-Everywhere Agreement [DPPU86]

- Byzantine agreement in partially connected networks
- Transmission scheme to simulate sending of a message between any two nodes
- If nodes $\notin W (= V - \mathcal{X}(T))$, then simulation is faithful
- $\Rightarrow$ Possible to simulate BA protocol for fully connected networks treating processors in $\mathcal{X}(T)$ as faulty (no privacy)
- “Almost-everywhere broadcast”
Almost-Everywhere Agreement (cont’d)

- [DPPU86] graphs:
  - Unbounded degree \( n^\varepsilon, 0 < \varepsilon < 1 \)
  - Bounded degree (butterfly, expander graphs)

- Objective: Large sets \( T \), “small” \( \chi(T) \)

- [Upf92]: Bounded-degree graphs with
  - \( T = O(n) \), \( \chi(T) = O(n) \)
  - Only one uncorrupted path between pairs of nodes in \( W \)
Admissible Graphs

Almost-everywhere \((i,t)\)-admissible graphs

1. Almost-everywhere broadcast in \(W\);
2. there exists a computable map \(\text{Select-Path}(G,u,v)\) s. t.
   
   \[
   \begin{align*}
   &\forall u,v \in V, \ |\text{PATHS}(u,v)| \in O(\text{poly}(n)) \\
   &\forall u,v \in W, \ \text{PATHS}(u,v) \text{ contains } \geq i \ \text{disjoint uncorrupted paths}
   \end{align*}
   \]
Lemma: Given two \((1,t)\)-admissible graphs \(G_n(V,E)\) and \(G'_{2n}(V',E')\), it is possible to construct a \((2,t)\)-admissible graph \(G''_{2n}(V'',E'')\) with \(|W''| = 2n - O(X'')\), where \(X'' = X + X'\).
X-MPC: Ingredients

- “Almost-everywhere agreement” [DPPU86]
  Players in W can implement a broadcast channel
  → Almost-everywhere \((i,t)\)-admissible graphs

- Secure message transmission (SMT) [DDWY89] by public discussion
  → Obtain pairwise secure channels between nodes in W

- Verifiable secret sharing (VSS) [CGMA85]
  → Implement Commit phase of X-MPC
Secret Sharing [Sha79, Bla79]

Less than $t + 1$ players have no info’ about the secret
Secret Sharing (cont’d)

Sharing Phase

\[ v_1 \quad v_2 \quad v_3 \quad \ldots \quad v_n \]

Reconstruction Phase

\[ \geq t + 1 \text{ players can reconstruct the secret} \]

Players are assumed to give their shares honestly.
**Verifiable Secret Sharing** [CGMA85]

- Extends secret sharing to the case of *active* corruptions (corrupted players, incl. Dealer, may not follow the protocol)
- *Adaptive* adversary
- **Reconstruction Phase:** Each player obtains

  \[ s' = \text{Rec}(v'_1, v'_2, \ldots, v'_n) \]
Verifiable Secret Sharing [CGMA85]

- Extends secret sharing to the case of \textit{active} corruptions (corrupted players, incl. Dealer, may not follow the protocol)
- \textit{Adaptive} adversary
- **Reconstruction Phase:** Each player obtains
  \[ s' = \text{Rec}(v'_1, v'_2, \ldots, v'_n) \]

\( n > 3t \) necessary and sufficient for VSS [BGW88], and there exist efficient protocols achieving it [GIKR02, FGGPS06]

**VSS network model:** \( p2p \) private channels + broadcast
Verifiable Secret Sharing [CGMA85]

- Extends secret sharing to the case of active corruptions (corrupted players, incl. Dealer, may not follow the protocol)
- Adaptive adversary
- Reconstruction Phase: Each player obtains
  \[ s' = \text{Rec}(v'_1, v'_2, \ldots, v'_{n}) \]

\( n > 3t \) necessary and sufficient for VSS [BGW88], and there exist efficient protocols achieving it [GIKR02, FGGPS06]

VSS network model: \textit{p2p private channels} + broadcast
X-MPC: Ingredients

- “Almost-everywhere agreement” [DPPU86]
  Players in W can implement a broadcast channel
  → Almost-everywhere i-admissible graphs

- Secure message transmission (SMT) [DDWY89]
  by public discussion
  → Obtain pair-wise secure channels between nodes in W

- Verifiable secret sharing (VSS) [CGMA85]
  → Implement Commit phase of X-MPC
Secure Message Transmission by Public Discussion

Sender $S$
message $m$

Receiver $R$

SMT [DDWY89]: $n$ channels, $t$ (actively) corrupted by $A$

Problem: Transmit $m$ privately and ($\varepsilon$-)reliably ($n > 2t+1$)
Secure Message Transmission by Public Discussion

Sender $S$

message $m$

Receiver $R$

SMT [DDWY89]: $n$ channels, $t$ (actively) corrupted by $A$

Problem: Transmit $m$ privately and ($\varepsilon$-)reliably ($n > 2t+1$)

Plus public channel: $n > t$ ($n > t+1$)
Protocol Pub-SMT

Send message $M$, $|M| = q$, $k = k(q, N, \varepsilon)$

1. $S \rightarrow R$: Send random $R_i$, $|R_i| = O(k)$ over each channel, $C_i$, $1 \leq i \leq N$

2. $S \rightarrow R$: Open $O(k)$ randomly chosen positions in $R_i$, $1 \leq i \leq N$
   (Call remaining string $R_i^*$)

3. $R \rightarrow S$: Identities of faulty channels
   ($N' \leq N$ : Non-faulty channels)

4. $S \rightarrow R$: $M = M_1 \oplus M_2 \oplus \ldots \oplus M_{N'}$
   Send $(M_i \oplus R_i^*)$, $1 \leq i \leq N'$
Protocol Pub-SMT

Send message $M$, $|M| = q$, $k = k(q, N, \varepsilon)$

1. $S \rightarrow R$: Send random $R_i$, $|R_i| = O(k)$ over each channel, $C_i$, $1 \leq i \leq N$

2. $S \rightarrow R$: Open $O(k)$ randomly chosen positions in $R_i$, $1 \leq i \leq N$
   (Call remaining string $R_i^*$)

3. $R \rightarrow S$: Identities of faulty channels
   ($N' \leq N$ : Non-faulty channels)

4. $S \rightarrow R$: $M = M_1 \oplus M_2 \oplus \ldots \oplus M_{N'}$
   
   Send $(M_i \oplus R_i^*)$, $1 \leq i \leq N'$

Theorem: Pub-SMT is a four-round SMT protocol transmitting

$O(\max(q, \log N/\varepsilon))$ bits on each of the $N$ channels and

$N \cdot O(\max(q, \log N/\varepsilon))$ bits over the public channel.
Talk Plan

- Secure multi-party computation (MPC)
- *Almost-everywhere* MPC (X-MPC)
- Related work
- Tools & ingredients
- X-MPC protocols
X-MPC Protocols

- Nodes in privileged set W have
  1. Almost-everywhere broadcast
  2. p2p private channels (Simulated by Pub-SMT)

- General strategy:
  - MPC on (2,t)-admissible graphs with T,X and W s.t. X < n/3
    replacing Sends & Receives of full MPC protocol by Pub-SMT
  - Communication structure: “super-round,” with players taking turns* (recall “rushing” adversary)

* For simplicity
X-MPC Protocols (cont’d)

Protocol C&C-MPC: Compute $F(x_1, \ldots, x_n)$

1. **Commit phase:** Sharing phase of VSS protocol. ($n$ executions are run.) At the end of the phase, player $P_i$ holds
   
   $$x^*_i = (v^1_i, \ldots, v^n_i)$$

2. **Computation phase:** Players execute original MPC protocol on “augmented” function
   
   $$F^*(x^*_1, \ldots, x^*_n) = F(\text{Rec}(v^1_1, \ldots, v^n_1), \ldots, \text{Rec}(v^1_n, \ldots, v^n_n))$$
X-MPC Protocols (cont’d)

Protocol C&C-MPC: Compute $F(x_1, \ldots, x_n)$

1. **Commit phase:** Sharing phase of VSS protocol. ($n$ executions are run.) At the end of the phase, player $P_i$ holds

   $$x^*_i = (v^1_i, \ldots, v^n_i)$$

2. **Computation phase:** Players execute original MPC protocol on “augmented” function

   $$F^*(x^*_1, \ldots, x^*_n) = F(\text{Rec}(v^1_1, \ldots, v^n_1), \ldots, \text{Rec}(v^1_n, \ldots, v^n_n))$$

**Theorem:** $G_n = (V,E)$, 2-admissible graph, $X < n/3$. Then C&C-MPC achieves X-MPC against adaptive $t$-adversary.
X-MPC on Classes of Networks

- $G_n$ of degree $O(n^\varepsilon)$, $t = O(n) \rightarrow O(t)$-MPC [DPPU86]

- $G_n$ of constant degree, $t = O(n/\log n) \rightarrow O(t)$-MPC [DPPU86]

- $G_n$ of constant degree, $t = O(n) \rightarrow O(t)$-MPC(**) [U92]
  (** Inefficient
Summary and Future Research

- Introduced *almost-everywhere MPC* (X-MPC), using
  1. AE (2,t)-admissible graphs
  2. SMT by public discussion
- Efficiency (e.g., [BG92] techniques)
- Security definitions: *Meaningful* simulation-based definition
- Pub-SMT: Comm. improvements and lower bounds
- Poly-time protocol for AE-agreement (and thus AE-MPC) on bounded-degree networks tolerating linear no. of corruptions
Almost-everywhere Secure Computation

Juan Garay  (Bell Labs)
Rafail Ostrovsky  (UCLA)