Almost-everywhere Secure Computation

Juan Garay (Bell Labs)
Rafail Ostrovsky (UCLA)

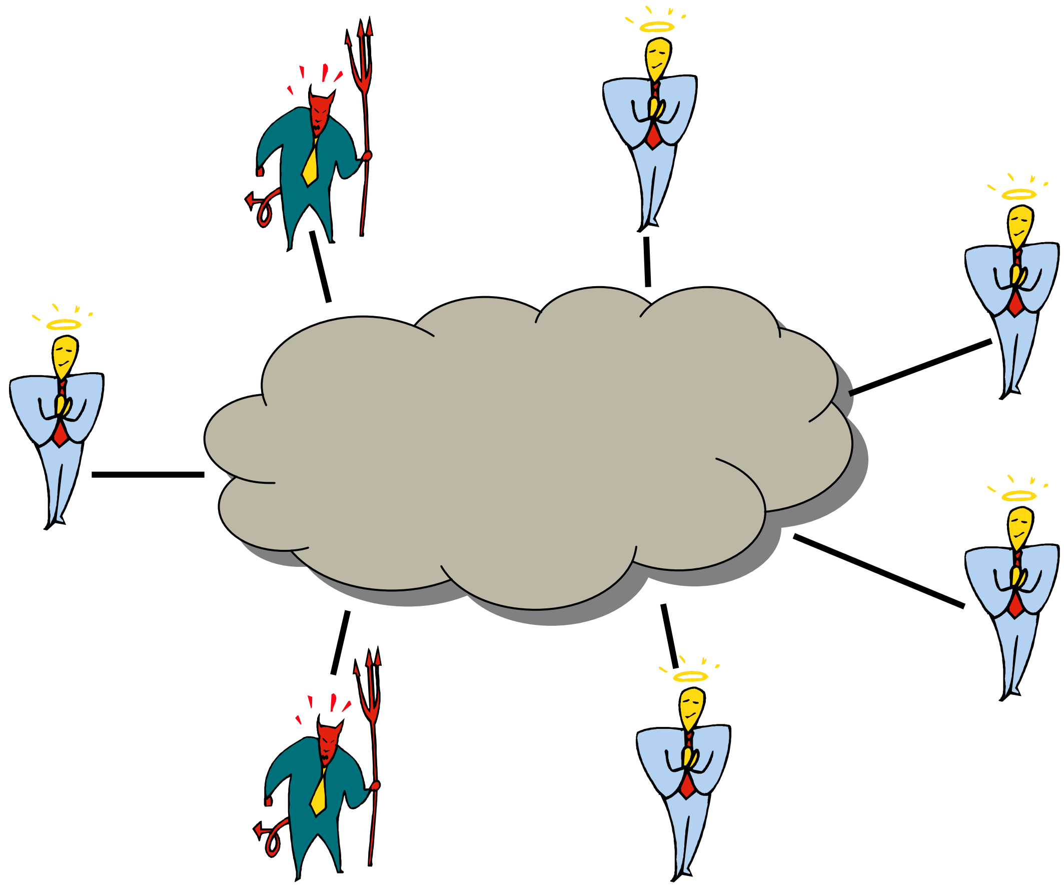
Secure Multi-party Computation (MPC)

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

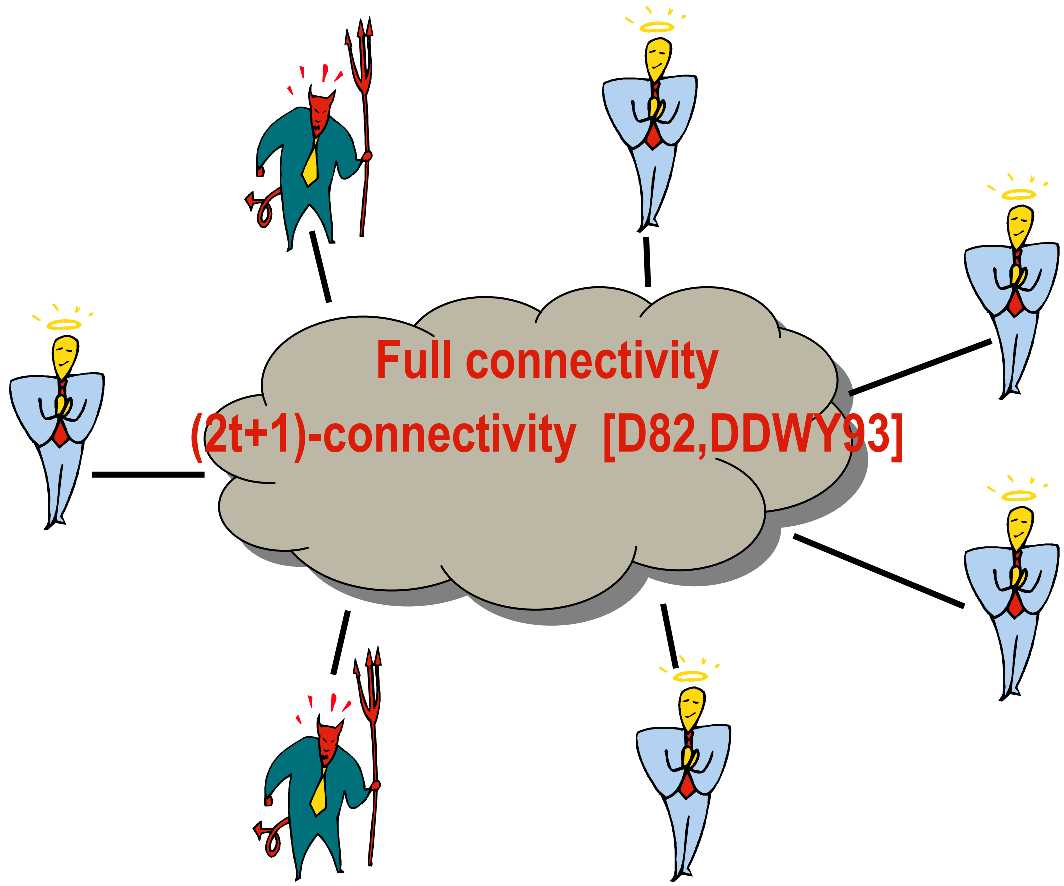


Unconditional (information-theoretic) MPC [BGW88, CCD88]:

n players, t corrupted, $n > 3t$

Alcatel·Lucent

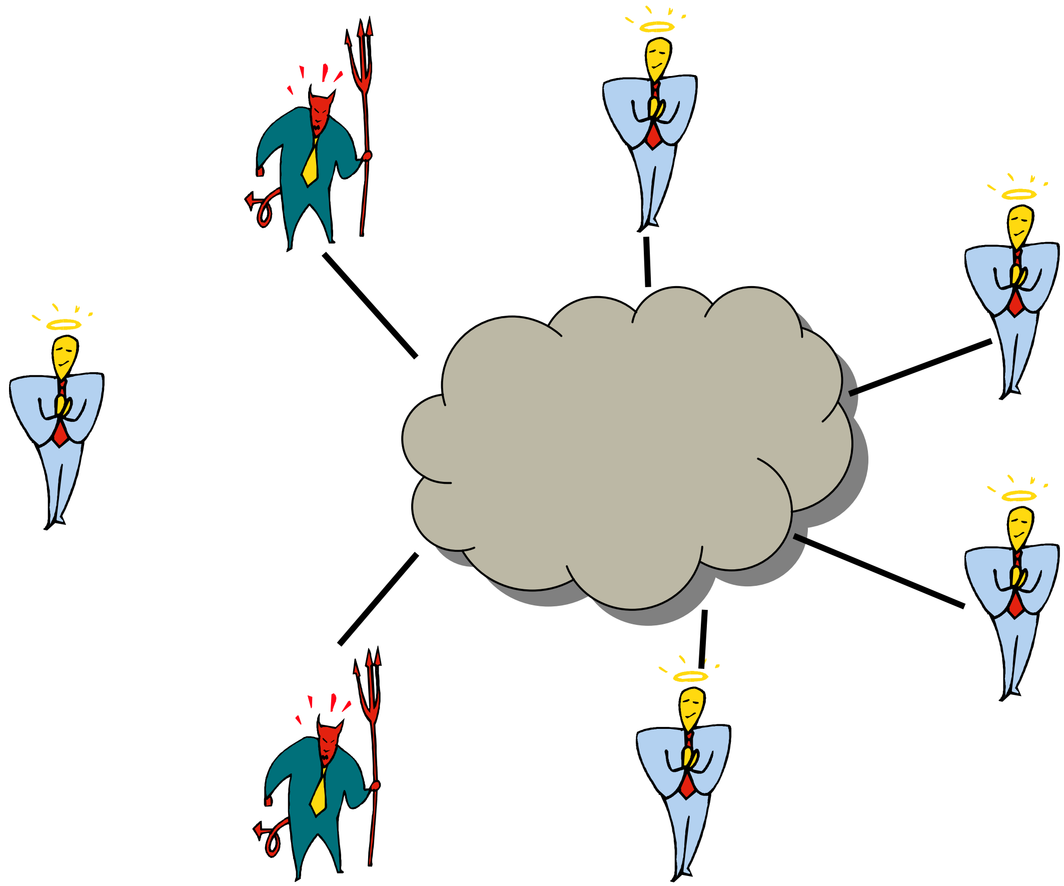
MPC: Network Requirements



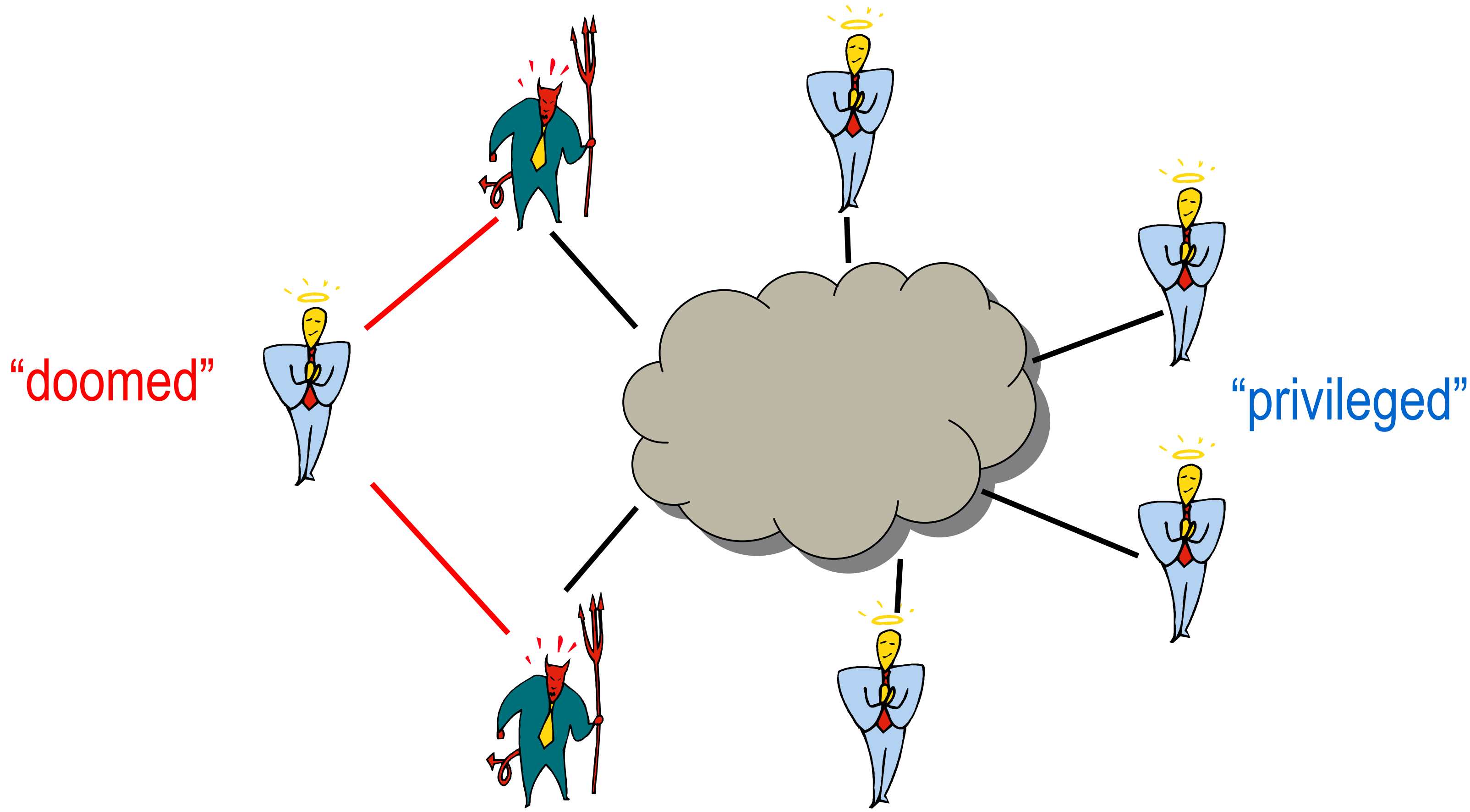
Unconditional (information-theoretic) MPC [BGW88, CCD88]:

n players, t corrupted, n > 3t

MPC on Incomplete Networks

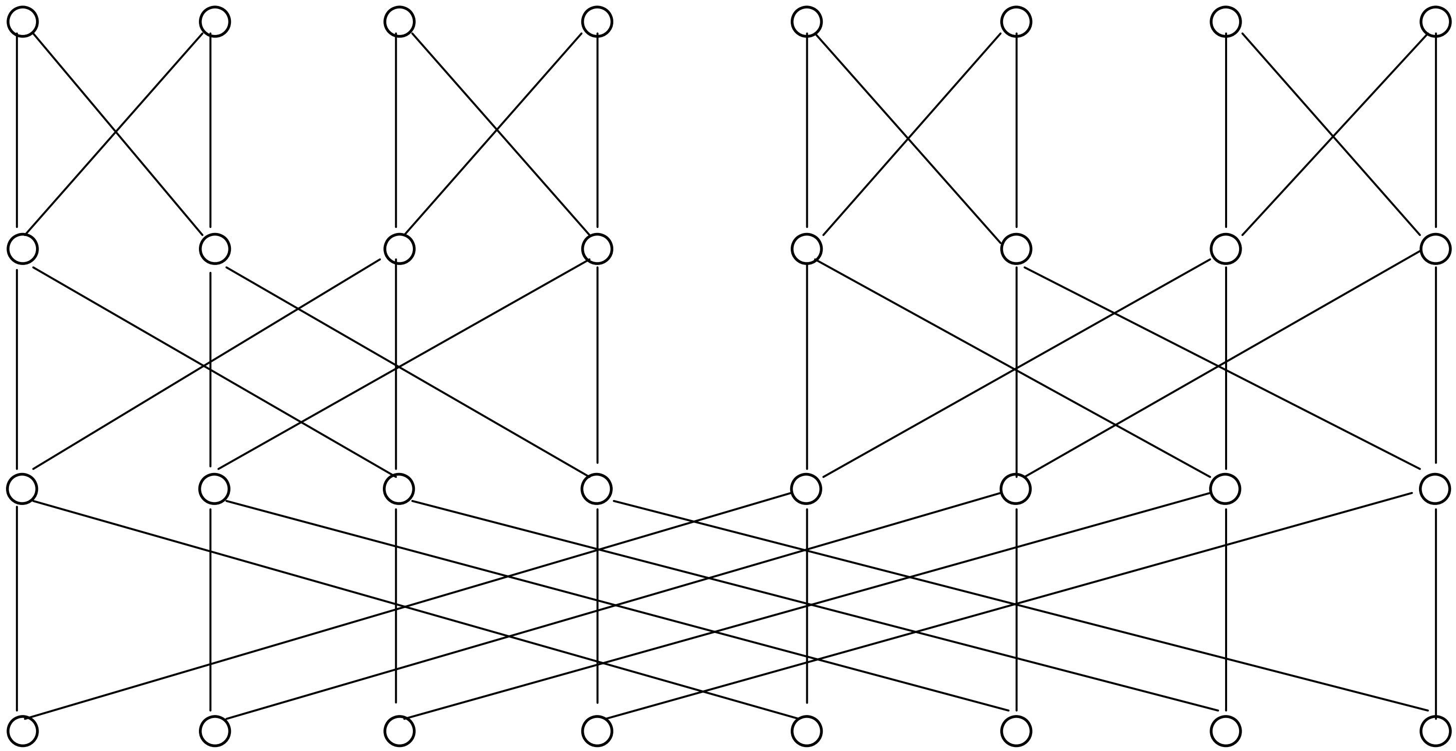


MPC on Incomplete Networks

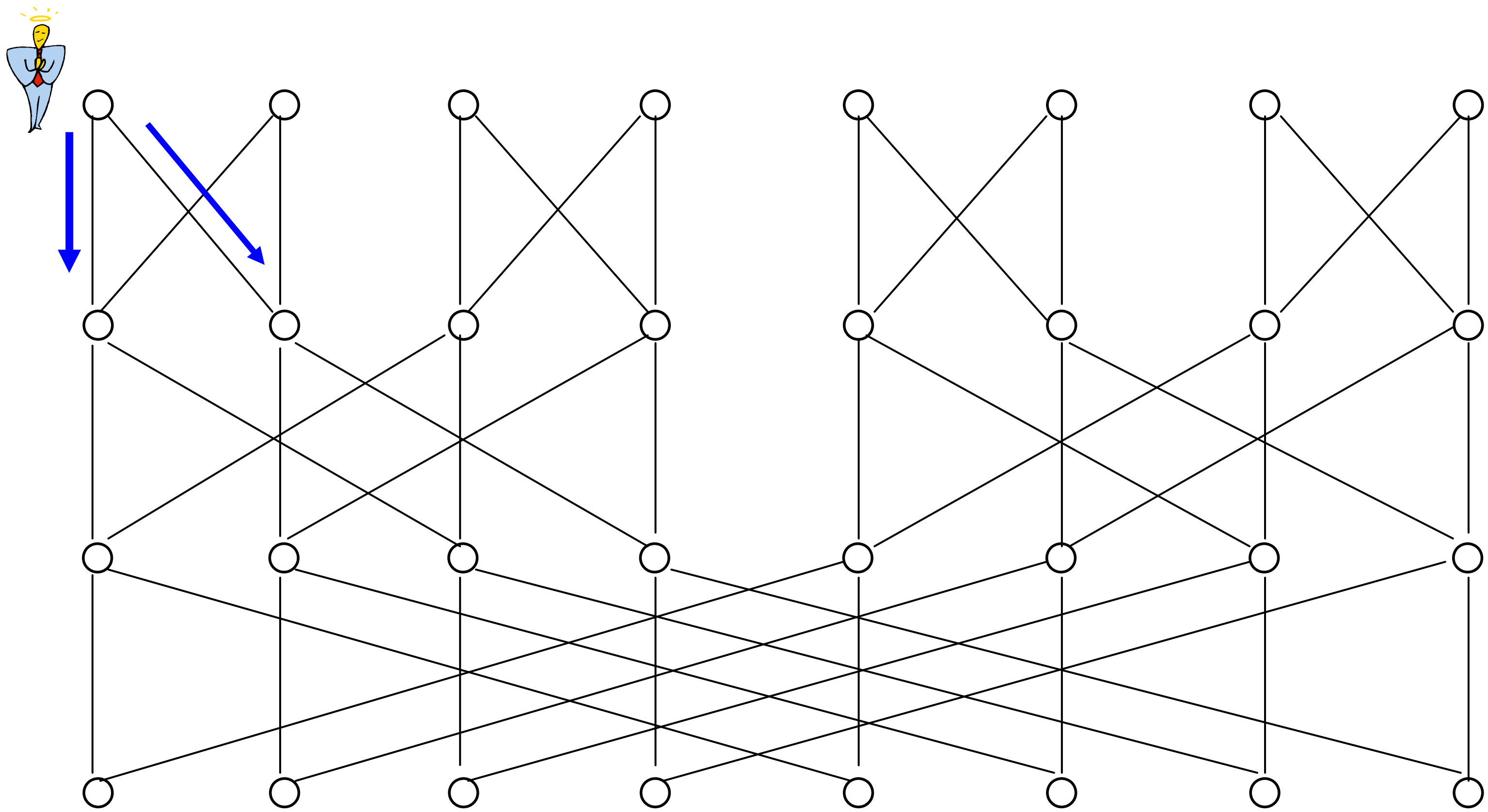


Small (constant) connectivity!

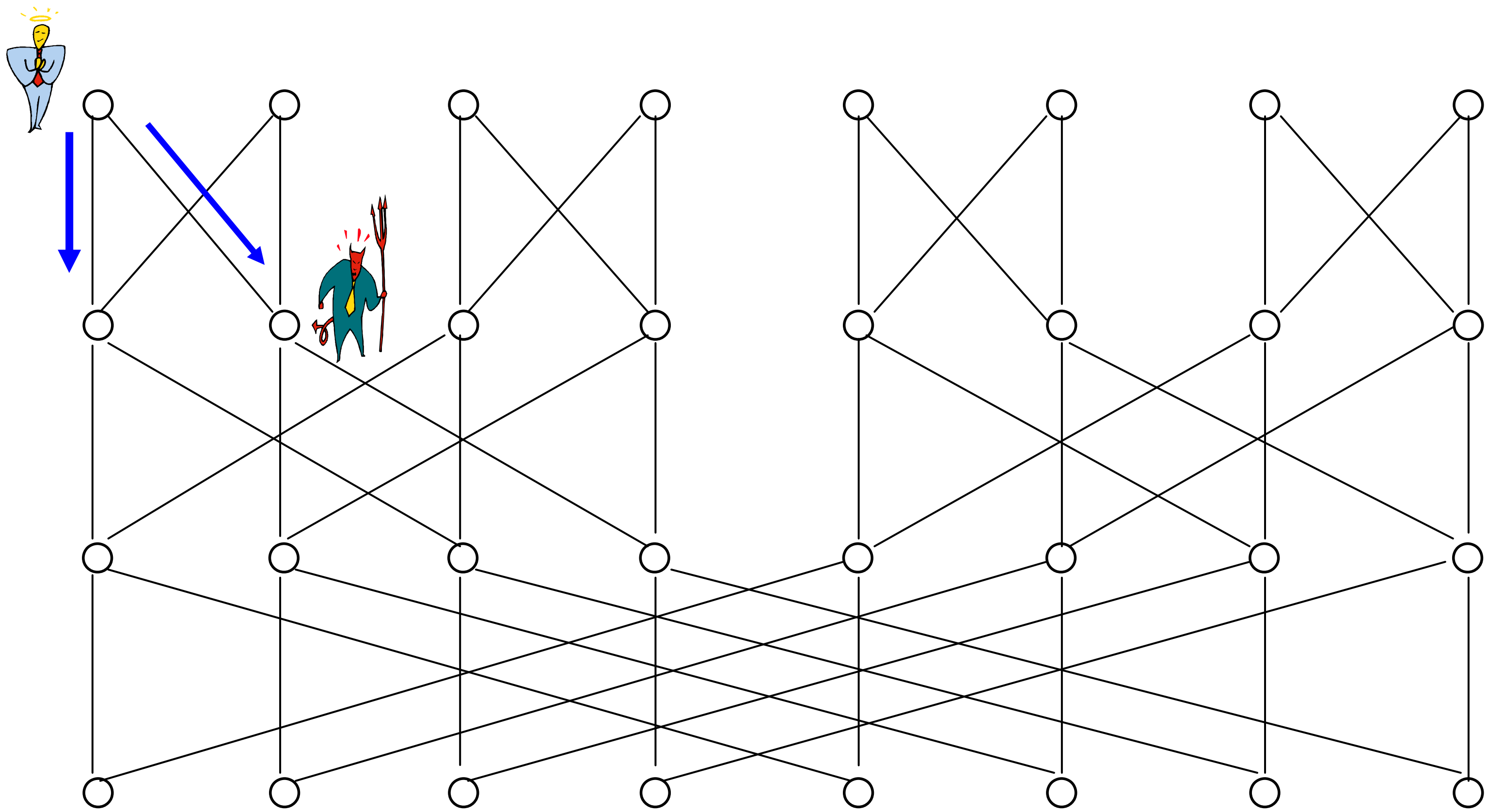
MPC on Incomplete Networks: Butterfly



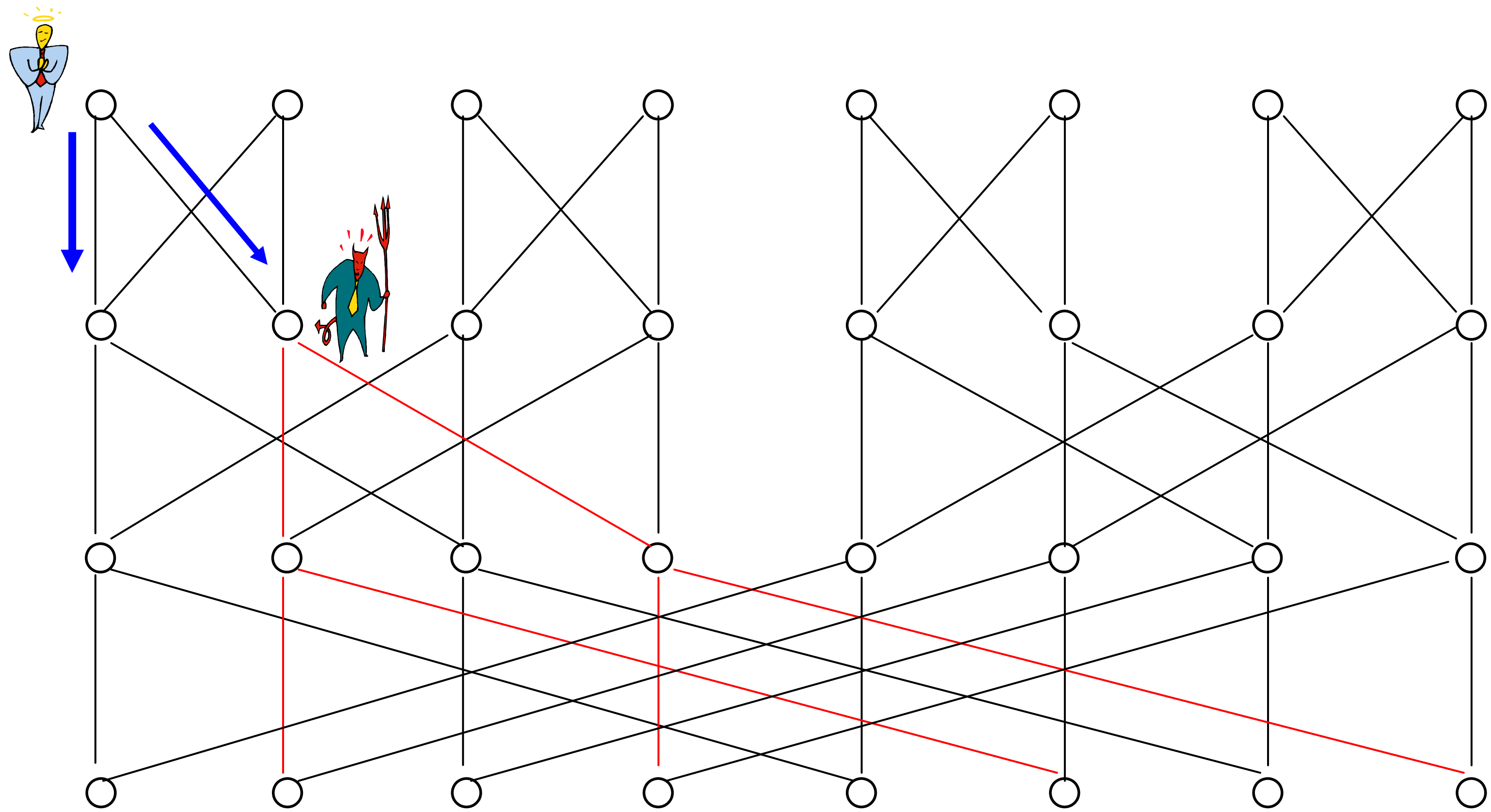
MPC on Incomplete Networks: Butterfly



MPC on Incomplete Networks: Butterfly

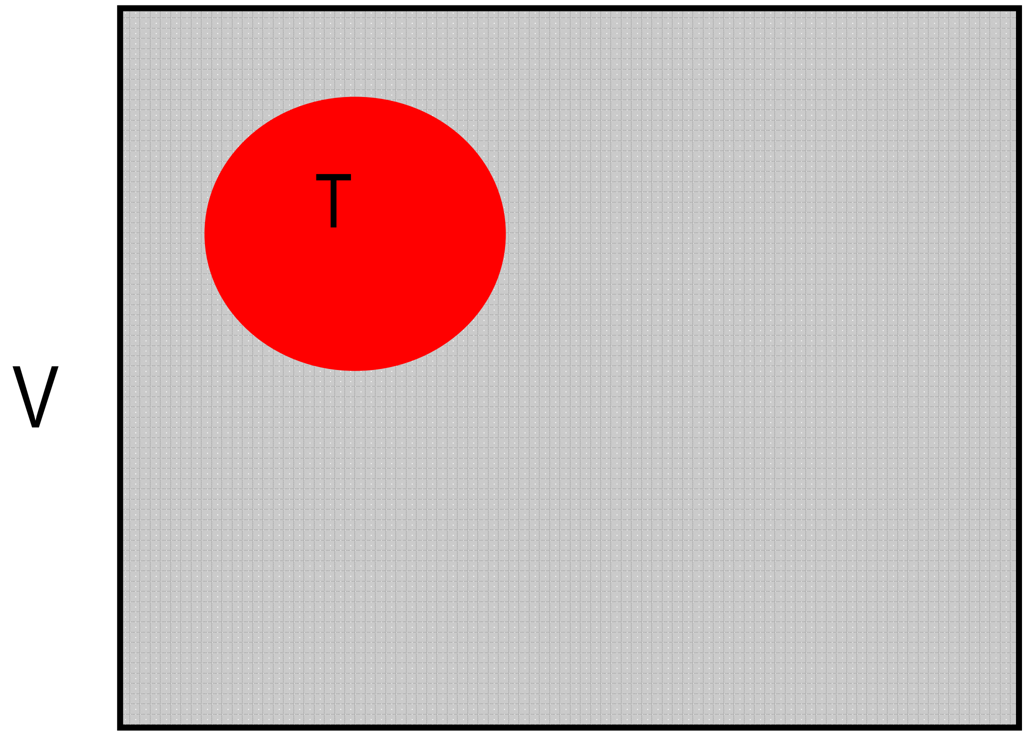


MPC on Incomplete Networks: Butterfly



This Work: *Almost-everywhere* MPC

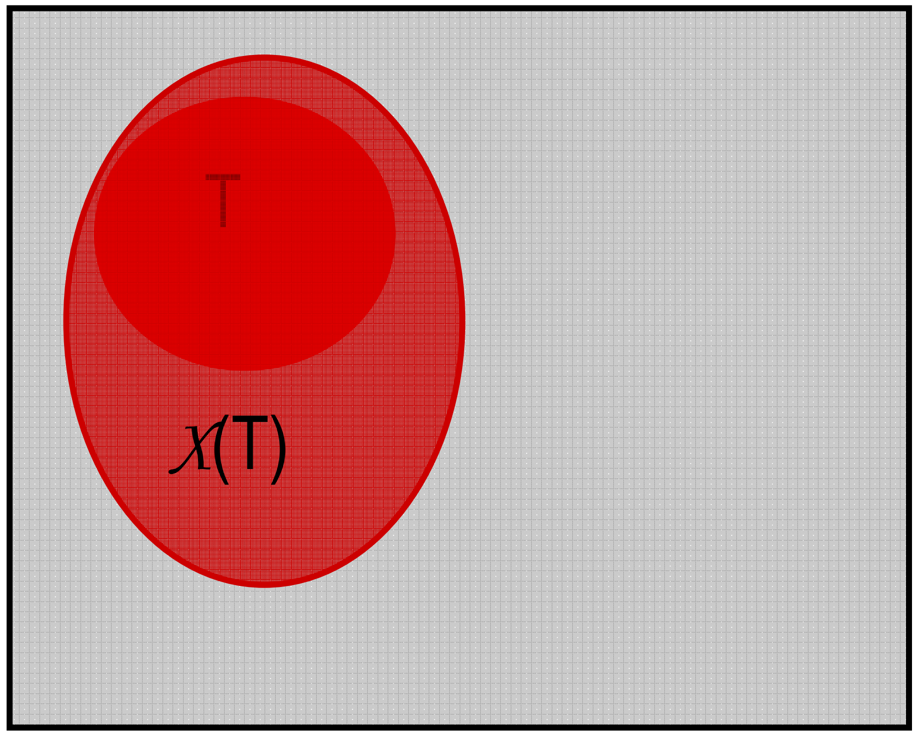
- "Give up" some of the players; guarantee security for a large fraction of them
- Adv. *implicitly* wiretaps by corrupting sufficiently many neighbors
- Capture **privacy** requirement
  - Definitional effort
  - *Adaptive* adversaries
- $G_n = (V,E)$



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
- $G_n = (V,E)$

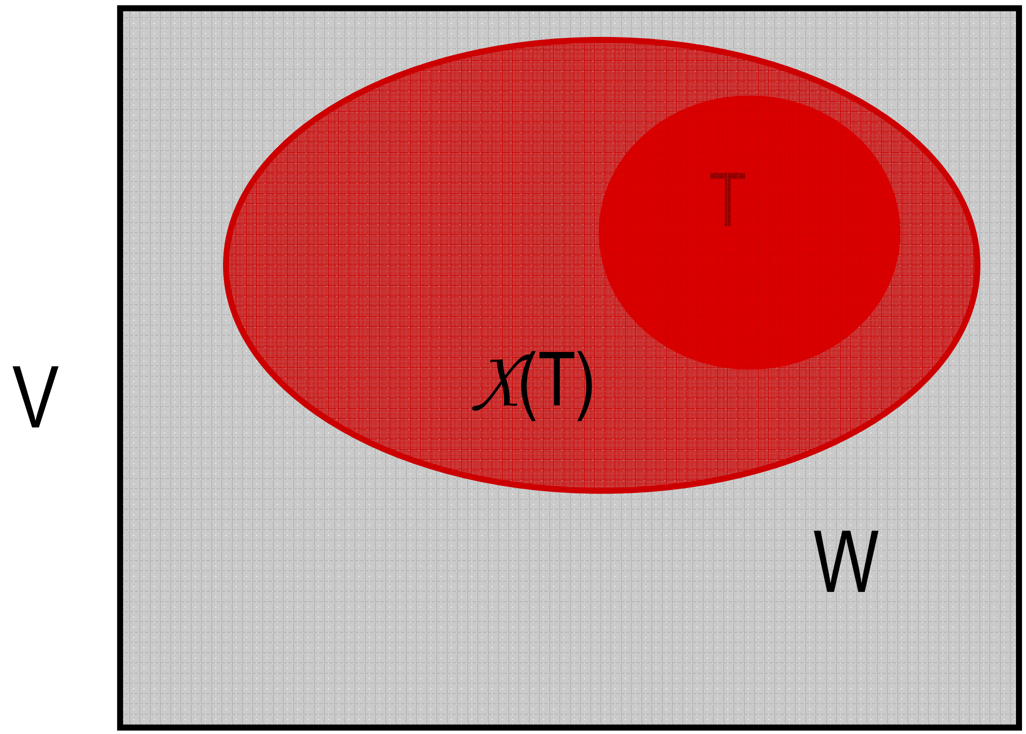
V



Alcatel·Lucent

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: *Privileged* set



Almost-everywhere MPC (cont'd)

- $G = (V,E)$, $|T| = t$, $\mathcal{P} = 2^V$
- $\mathcal{X}: \mathcal{P}^{(\leq t)} \rightarrow \mathcal{P}$
  1. $T_1 \subseteq T_2 \Rightarrow \mathcal{X}(T_1) \subseteq \mathcal{X}(T_2)$
  2. $T \subseteq \mathcal{X}(T)$

  $X = \max_T \{|\mathcal{X}(T)|\}$
- Protocol Π achieves **X-MPC** if $\exists W, |W| \geqq n - X$, s.t. all players in W are able to perform MPC
- Fully connected network: $\mathcal{X}(T) = T$

"Commit-and-Compute" Paradigm

A two-phase protocol Π 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 $X^*_W, Y^*_W, Z^*_{\mathcal{X}(T)}$ such that

$$F(X^*_W, Z^*_{\mathcal{X}(T)}) = F(Y^*_W, Z^*_{\mathcal{X}(T)})$$

   the adversary can't distinguish $\Pi(X^*_W, Z^*_{\mathcal{X}(T)})$ from $\Pi(Y^*_W, Z^*_{\mathcal{X}(T)})$

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

  → *Almost-everywhere (i,t)-admissible graphs*

- Secure message transmission (SMT) [DDWY89]

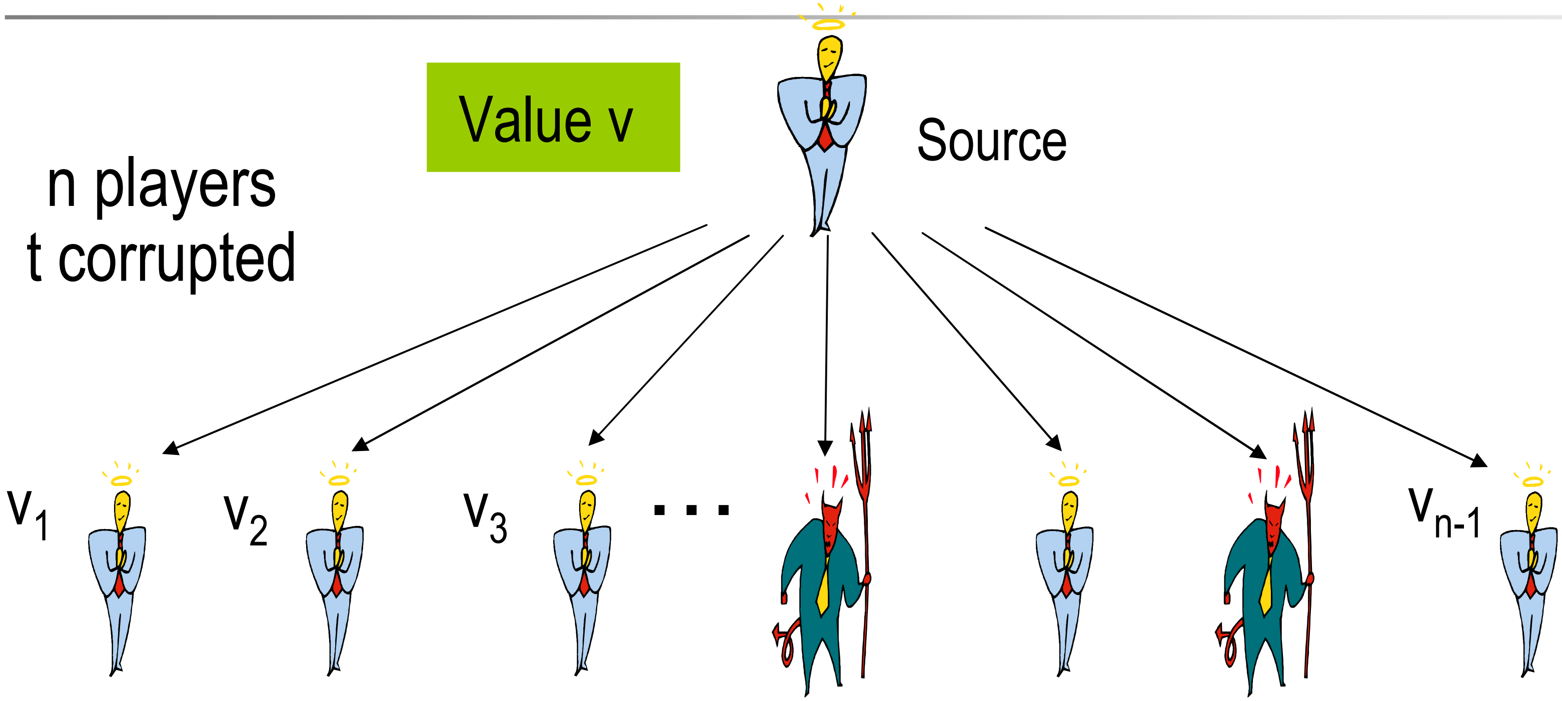
  by public discussion

  → Obtain pair-wise *secure* channels between nodes in W

- Verifiable secret sharing (VSS) [CGMA85]

  → Implement Commit phase of X-MPC

Broadcast (aka Byz. agreement) [PSL80, LSP82]



- 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 Є W (= V – $\mathcal{X}$(T)), then simulation is faithful
- ⇒ 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)

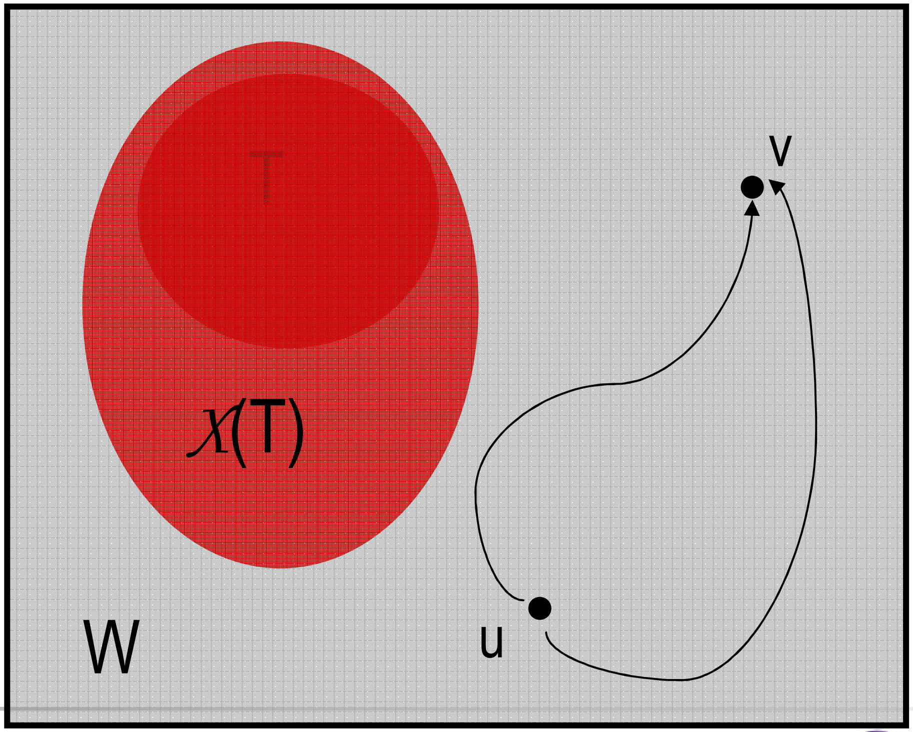
- [DPPU86] graphs:
  - Unbounded degree (n^{ε}, $0 < \varepsilon < 1$)
  - Bounded degree (butterfly, expander graphs)
- Objective: Large sets T, "small" $\mathcal{X}(T)$
- [Upf92]: Bounded-degree graphs with
  - $T = O(n)$, $\mathcal{X}(T) = O(n)$
  - Only one uncorrupted path between pairs of nodes in W

Admissible Graphs

Almost-everywhere (i,t)-admissible graphs

1. Almost-everywhere broadcast in W;
2. there exists a computable map Select-Path(G,u,v) s. t.
   - ∀ u,v ∈ V, |PATHS(u,v)| ∈ O(poly(n))
   - ∀ u,v ∈ W, PATHS(u,v) contains ≧ i disjoint uncorrupted paths

(2,t)-admissible graph V



Admissible Graphs (cont'd)

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]

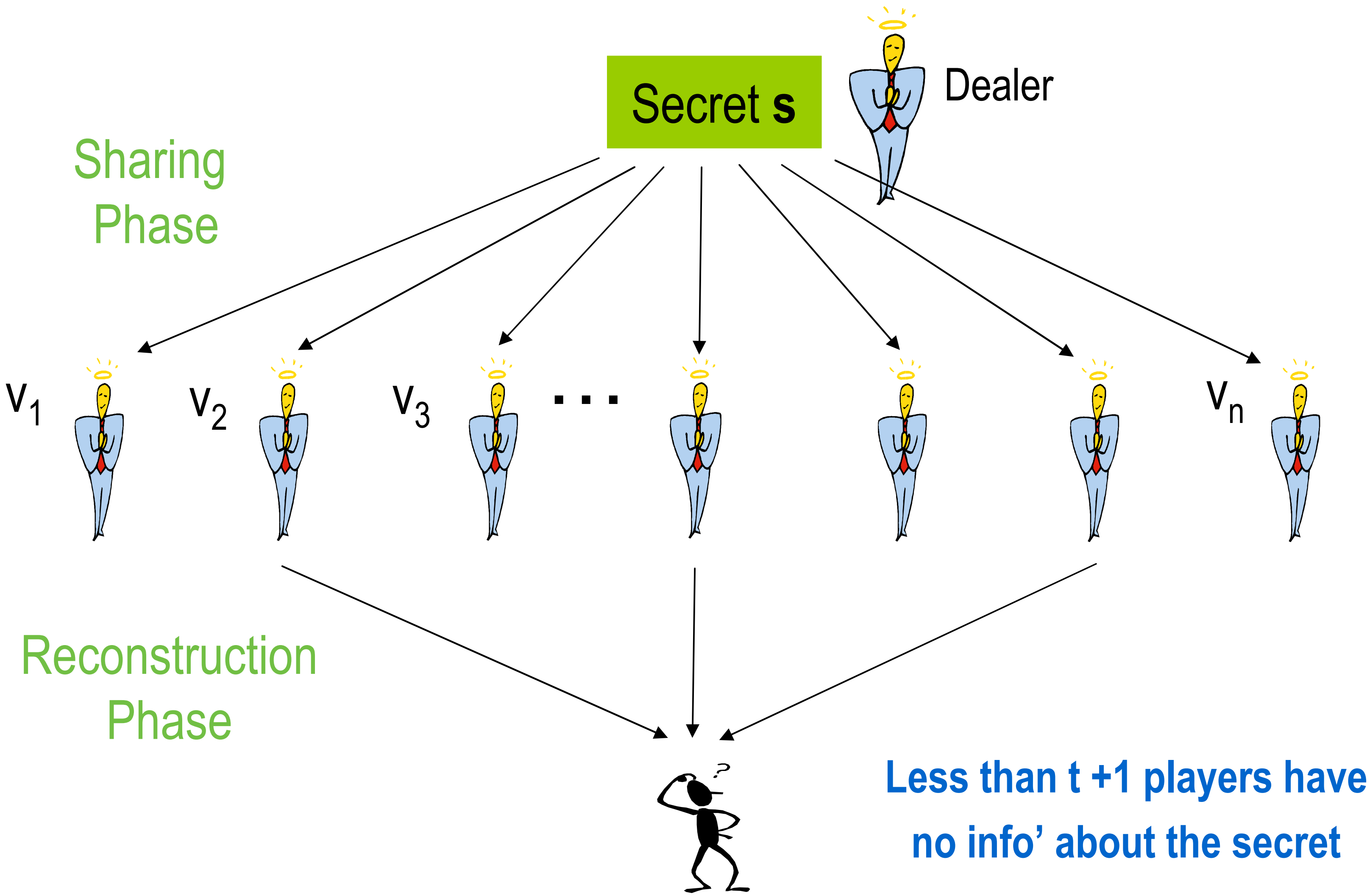
  by public discussion

  → Obtain pairwise *secure* channels between nodes in W

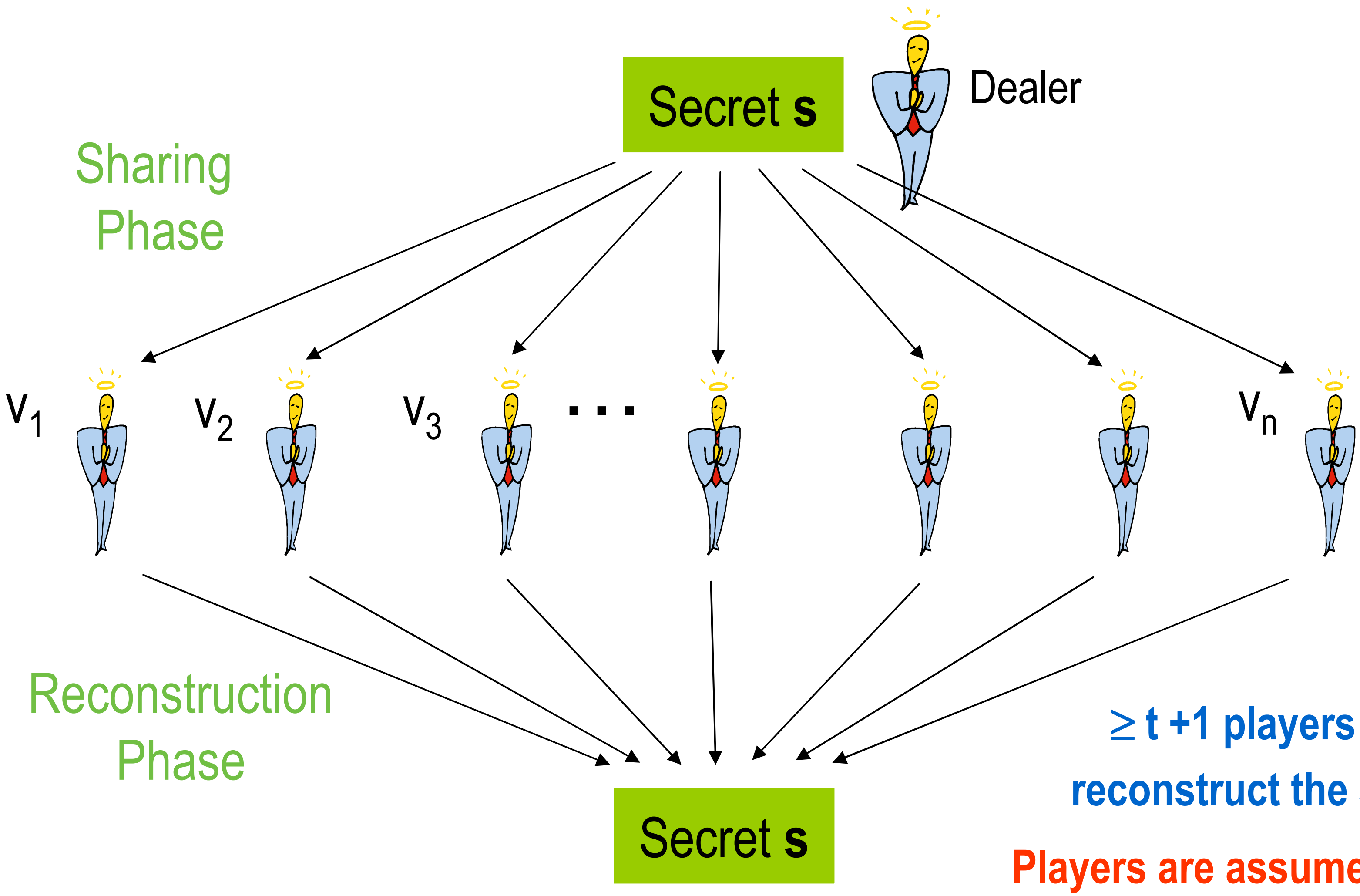
- Verifiable secret sharing (VSS) [CGMA85]

  → Implement Commit phase of X-MPC

Secret Sharing [Sha79, Bla79]



Secret Sharing (cont'd)



$\geq$ t +1 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

$$\mathbf{s'} = \mathrm{Rec}(v'_1, v'_2, \ldots, v'_n)$$

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

$$\mathbf{s'} = \mathrm{Rec}(v'_1, v'_2, \ldots, v'_n)$$

$n > 3t$ 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

$$s' = \mathrm{Rec}(v'_1, v'_2, \ldots, v'_n)$$

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

VSS network model: p2p *private* channels + broadcast

X-MPC: Ingredients

- "Almost-everywhere agreement" [DPPU86]

  Players in W can implement a broadcast channel

  → *Almost-everywhere i-admissible graphs*

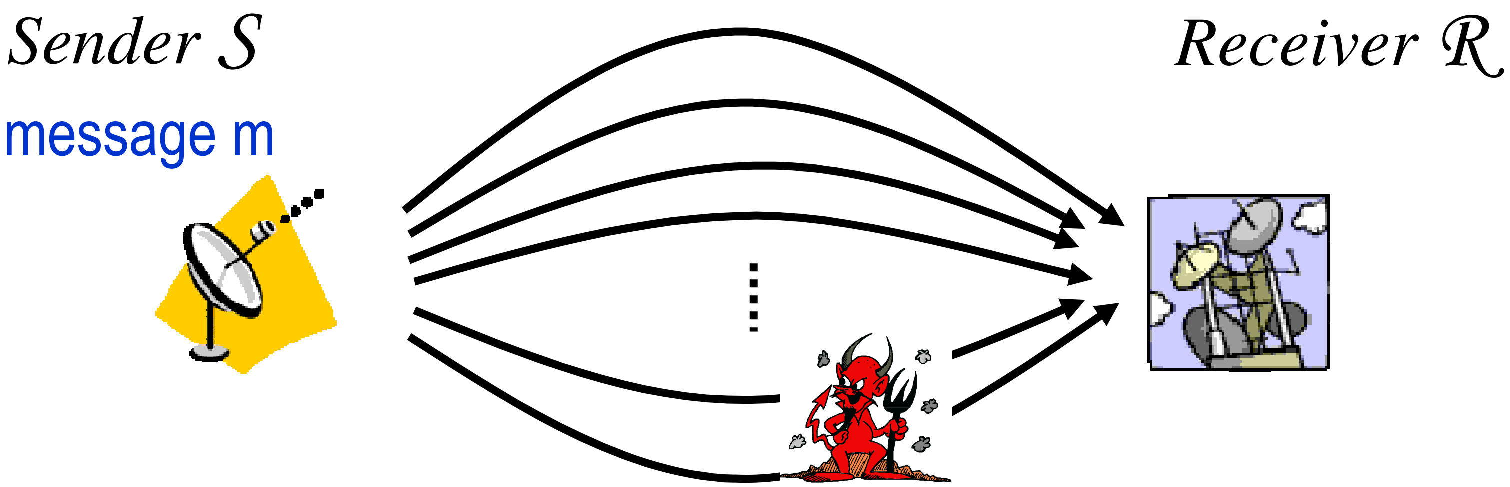
- Secure message transmission (SMT) [DDWY89] *by public discussion*

  → Obtain pair-wise *secure* channels between nodes in W

- Verifiable secret sharing (VSS) [CGMA85]

  → Implement Commit phase of X-MPC

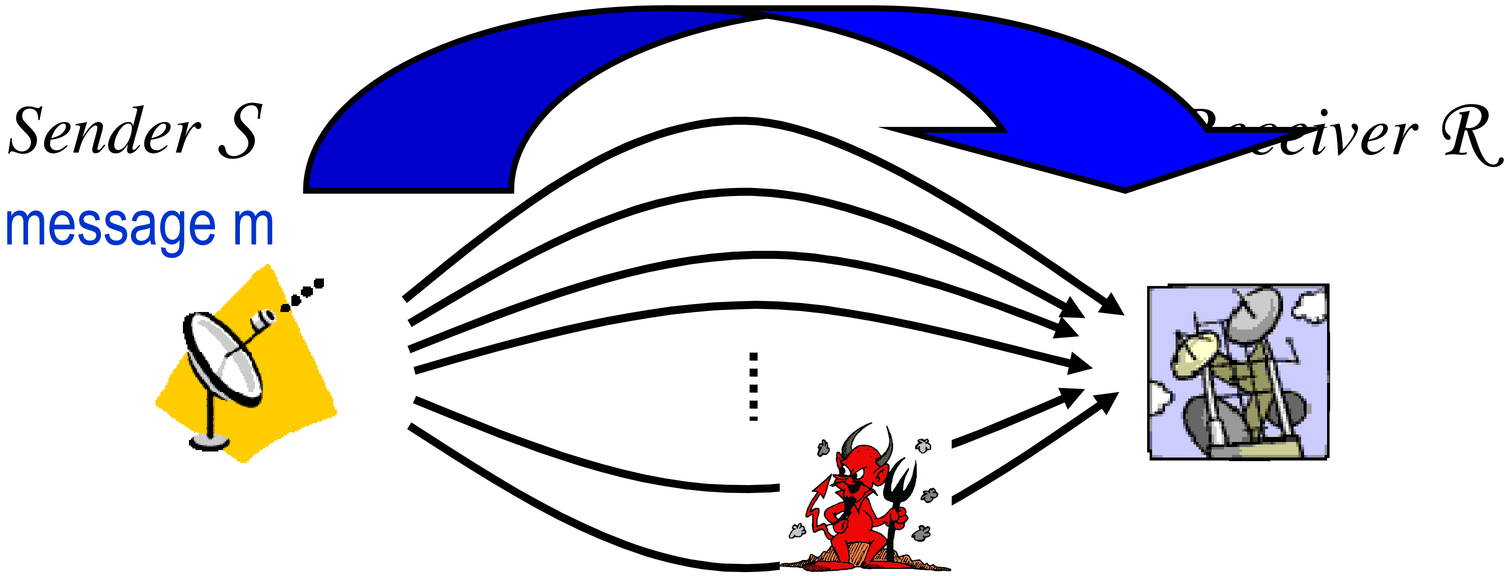
Secure Message Transmission *by Public Discussion*



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

Problem: Transmit m *privately* and (ε-)*reliably* (n > 2t+1)

Secure Message Transmission *by Public Discussion*



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

Problem: Transmit m *privately* and (ε-)*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. $\mathcal{S} \rightarrow \mathcal{R}$: Send random R_i , $|R_i| = O(k)$ over each channel, C_i, $1 \leqq i \leqq N$
2. $\mathcal{S} \rightarrow \mathcal{R}$: Open $O(k)$ randomly chosen positions in R_i , $1 \leqq I \leqq N$
   (Call remaining string R_i^*)
3. $\mathcal{R} \rightarrow \mathcal{S}$: Identities of faulty channels
   ($N' \leqq N$: Non-faulty channels)
4. $\mathcal{S} \rightarrow \mathcal{R}$: $M = M_1 \oplus M_2 \oplus \ldots \oplus M_{N'}$
   Send $(M_i \oplus R_i^*)$, $1 \leqq I \leqq N'$

Protocol Pub-SMT

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

1. $\mathcal{S} \rightarrow \mathcal{R}$: Send random R_i , $|R_i| = O(k)$ over each channel, C_i, $1 \leqq i \leqq N$
2. $\mathcal{S} \rightarrow \mathcal{R}$: Open O(k) randomly chosen positions in R_i , $1 \leqq I \leqq N$

   (Call remaining string R_i^*)
3. $\mathcal{R} \rightarrow \mathcal{S}$: Identities of faulty channels

   ($N' \leqq N$: Non-faulty channels)
4. $\mathcal{S} \rightarrow \mathcal{R}$: $M = M_1 \oplus M_2 \oplus \ldots \oplus M_{N'}$

   Send $(M_i \oplus R_i^*)$, $1 \leqq I \leqq N'$

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

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

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

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

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 $O(n^{\varepsilon})$, $t = O(n)$ $\longrightarrow$ $O(t)$-MPC [DPPU86]
- G_n of *constant* degree, $t = O(n/\log n)$ $\longrightarrow$ $O(t)$-MPC [DPPU86]
- G_n of *constant* degree, $t = O(n)$ $\longrightarrow$ $O(t)$-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)