Teoria dei Giochi e Multi-Party Computation

Francesco Mucci

Scuola di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea in Informatica

Anno Accademico 2014-2015

UNIVERSITÀ
DEGLI STUDI
FIRENZE

Obbiettivo e motivazioni della tesi

$$\left.\begin{array}{r} \textit{Multi-Party Computation (MPC)} \\ \textit{Teoria dei Giochi} \end{array}\right\} \quad \textit{Gioco di Multi-Party Computation}$$

Obbiettivo Gioco di Multi-Party Computation

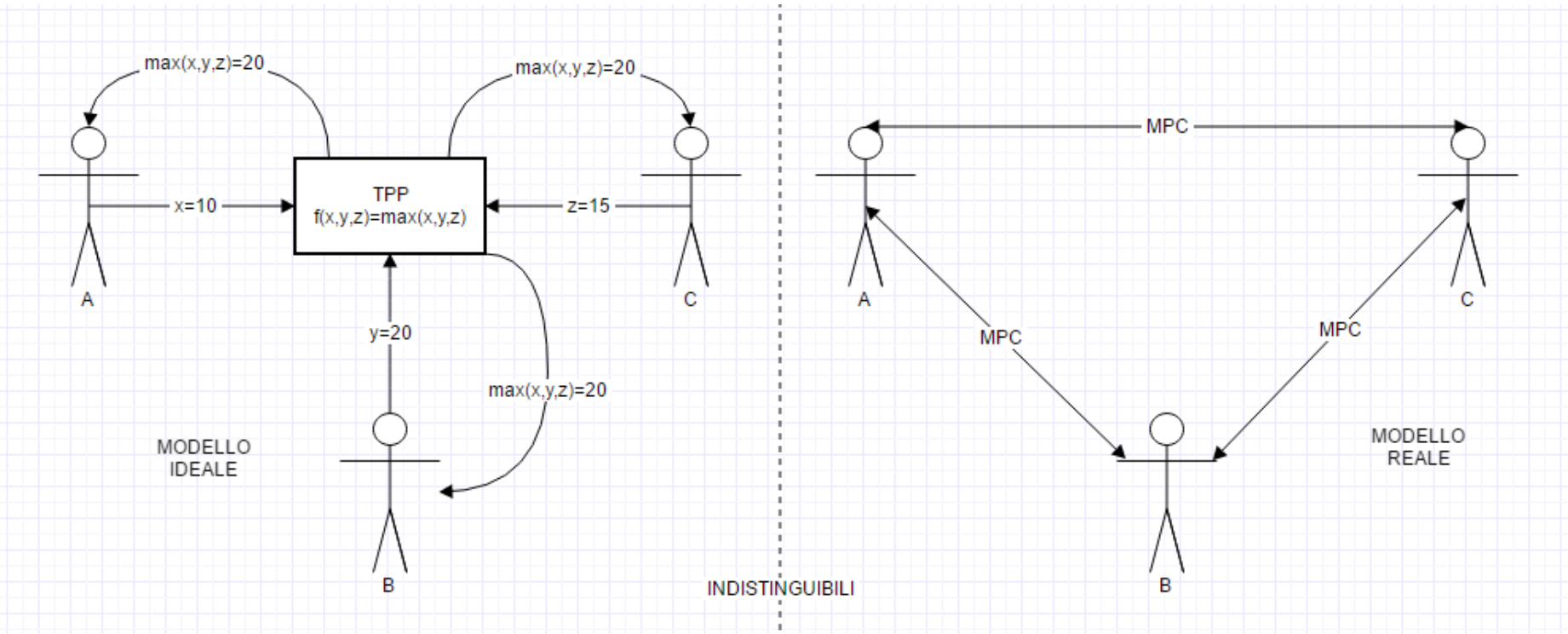
Comprendere in quali situazioni partecipanti razionali ad un protocollo MPC sono incentivati ad agire correttamente.

Motivazioni lavoro di tesi

Illustrare alcuni tra i più recenti risultati nel campo della Game-Theoretic MPC, rendendo l'argomento accessibile tramite una semplice ed essenziale introduzione alla *Multi-Party Computation* ed alla *Teoria dei Giochi*.

Problema di Multi-Party Computation sicura

- $P_1, \ldots, P_n$ soggetti; $\forall P_i$, t_i input segreto.
- $s = f(t_1, \ldots, t_n)$ funzione da calcolare.
- $\forall P_i : (t_i, s) \rightsquigarrow P_i$
- *Input privacy* e *Correttezza della computazione*.

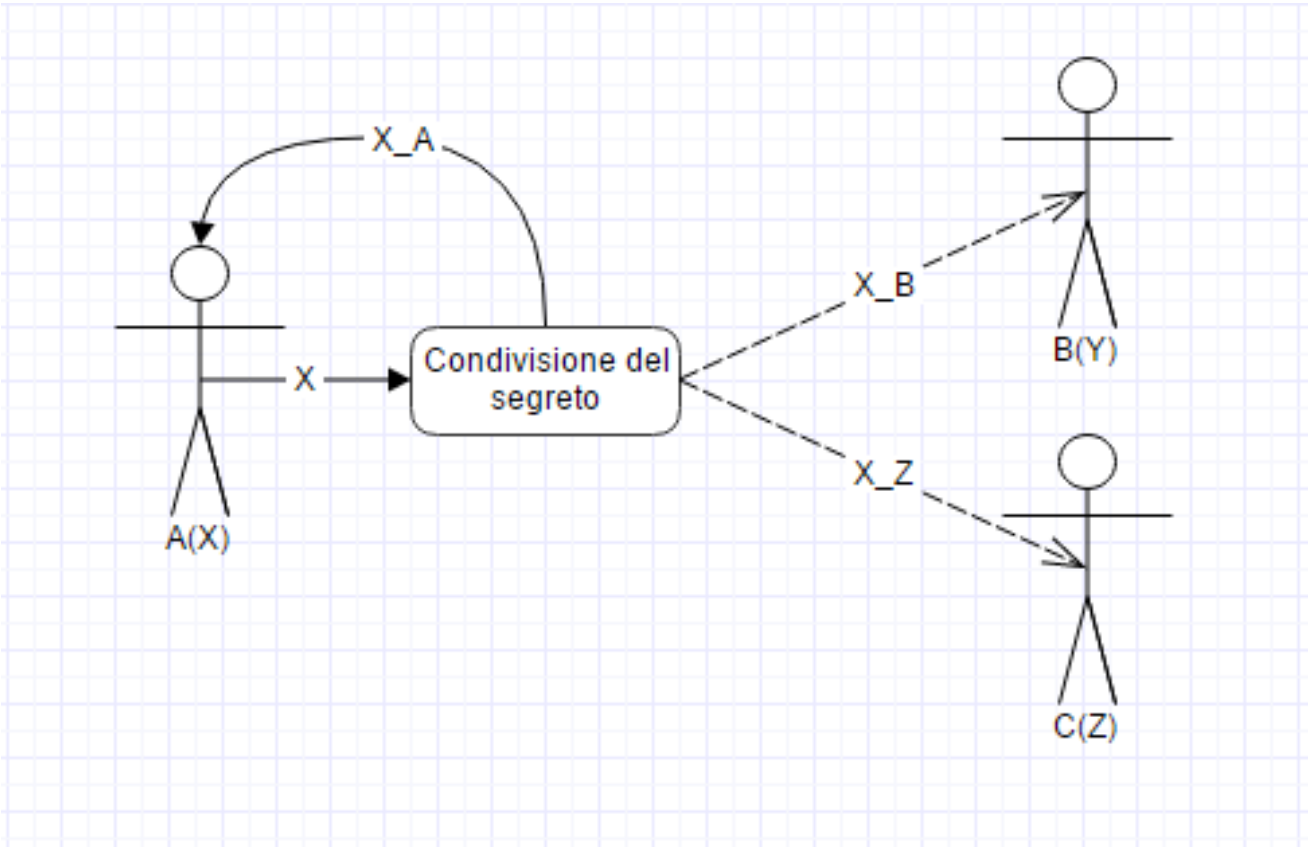


Struttura generale di un protocollo MPC

1. Condivisione degli input.
2. Computazione.
3. Rivelazione dell'output.

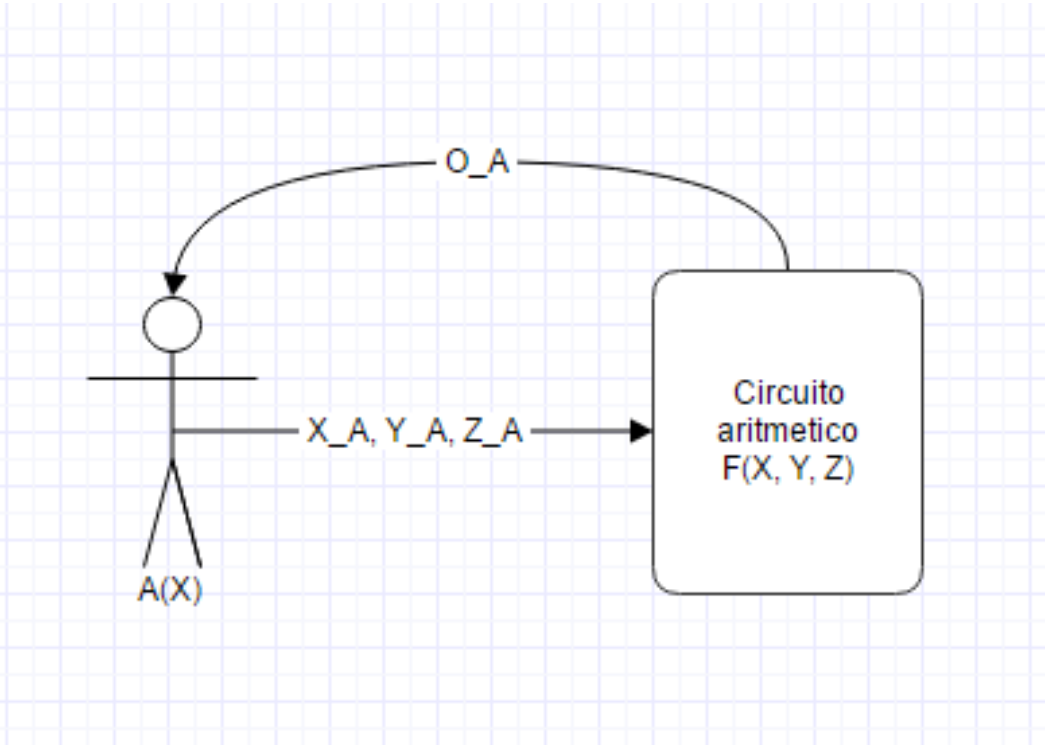
Struttura generale di un protocollo MPC

1. Condivisione degli input.
2. Computazione.
3. Rivelazione dell'output.



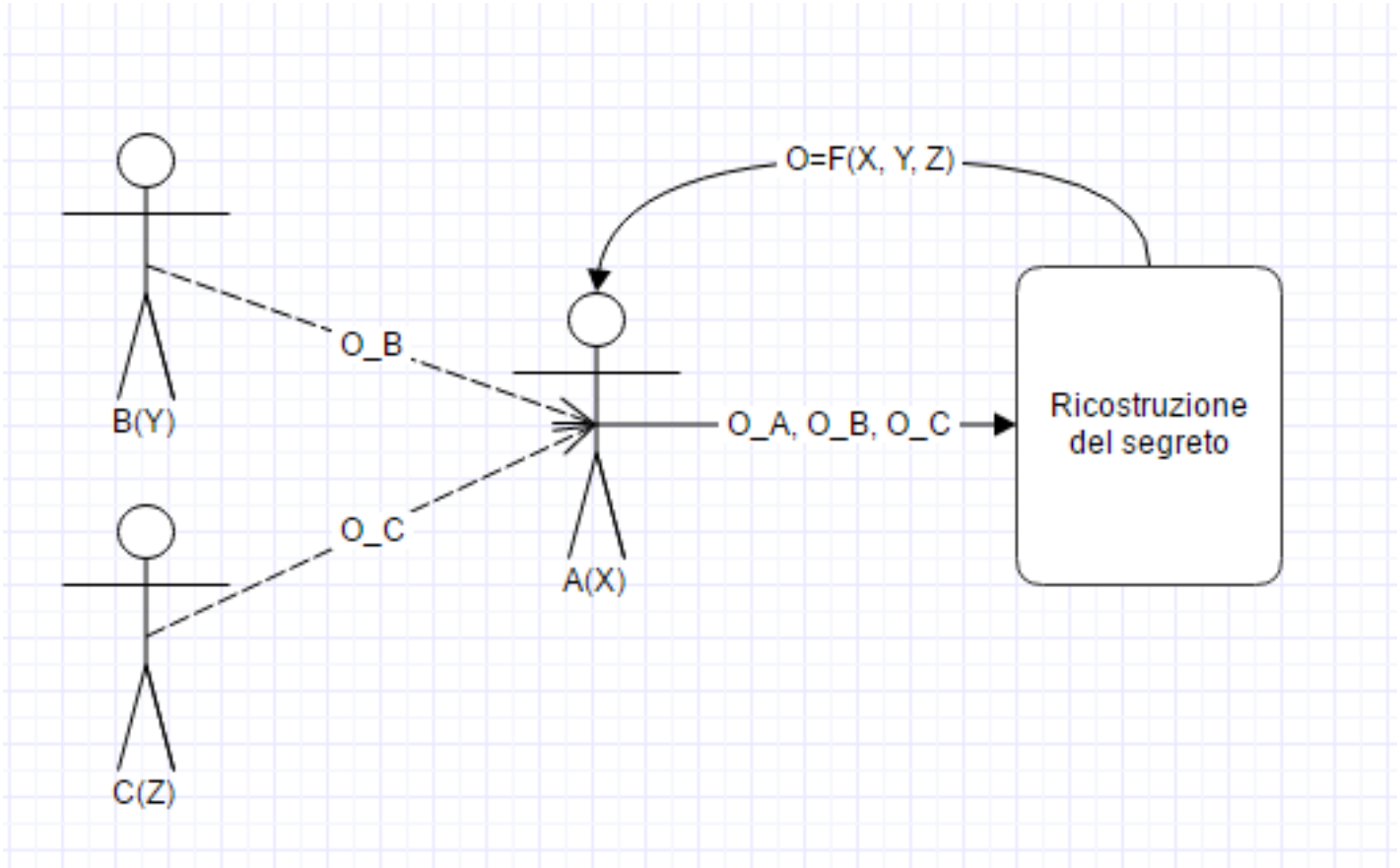
Struttura generale di un protocollo MPC

1. Condivisione degli input.
2. Computazione.
3. Rivelazione dell'output.



Struttura generale di un protocollo MPC

1. Condivisione degli input.
2. Computazione.
3. Rivelazione dell'output.



Concetti di Teoria dei Giochi

Teoria dei Giochi

Studia i modelli matematici (*giochi*) che descrivono l'interazione strategica, conflittuale e/o cooperativa, tra più soggetti intelligenti, razionali e tra loro indipendenti (*giocatori*).

Gioco in forma normale

Tupla $G = (N, A, u)$ dove:

- $N = P_1, \ldots, P_n$ insieme di $n \in \mathbb{N}$ *giocatori*;
- $A = A_1 \times \cdots \times A_n$; A_i è l'insieme delle possibili *azioni* di P_i.
- $u = (u_1, \ldots, u_n)$, con $u_i : A \to \mathbb{R}$ *funzione utilità* per P_i.

- $a = (a_1, \ldots, a_n)$ *profilo di azioni (risultato)*, con $a_i \in A_i$;
- $s_i \in S_i$ strategia: determina l'azione o le azioni che P_i eseguirà in ogni stadio del gioco per ogni possibile "storia". Algoritmo per giocare il gioco.
- $u_i(a)$ payoff di P_i per il dato risultato;

Gioco in forma normale

Tupla $G = (N, A, u)$ dove:

- $N = P_1, \ldots, P_n$ insieme di $n \in \mathbb{N}$ *giocatori*;
- $A = A_1 \times \cdots \times A_n$; A_i è l'insieme delle possibili *azioni* di P_i.
- $u = (u_1, \ldots, u_n)$, con $u_i : A \to \mathbb{R}$ *funzione utilità* per P_i.

Capacità giocatore

- Intelligente: dati due risultati è in grado di decidere quale sia migliore.
- Razionale: capacità logica di saper riconoscere le azioni per massimizzare i propri payoff.

Esempio: "Dilemma del prigioniero"

- Gioco one-shot ad azione simultanea
- Gioco ad informazione completa

		Sospettato-2	
		Confessa	Non confessa
Sospettato-1	Confessa	-3, -3	0, -4
	Non confessa	-4, 0	-1, -1

Esempio: "Dilemma del prigioniero"

		Sospettato-2	
		Confessa	Non confessa
Sospettato-1	Confessa	-3, -3	0, -4
	Non confessa	-4, 0	-1, -1

Equilibrio di Nash a strategie pure

- La *miglior risposta* di P_i al profilo $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ è una strategia pura $a_i^* \in A_i$ tale che $u_i(a_i^*, a_{-i}) \geq u_i(a_i, a_{-i}) \; \forall a_i \in A_i$.
- Un profilo strategico $a = (a_1, \ldots, a_n)$ è un *equilibrio di Nash* se, $\forall \; P_i$, a_i è la *migliore risposta* al profilo a_{-i}.

Perché analizzare la MPC usando la Teoria dei Giochi?

- partecipanti onesti o disonesti vs partecipanti razionali.

Evoluzione protocolli MPC

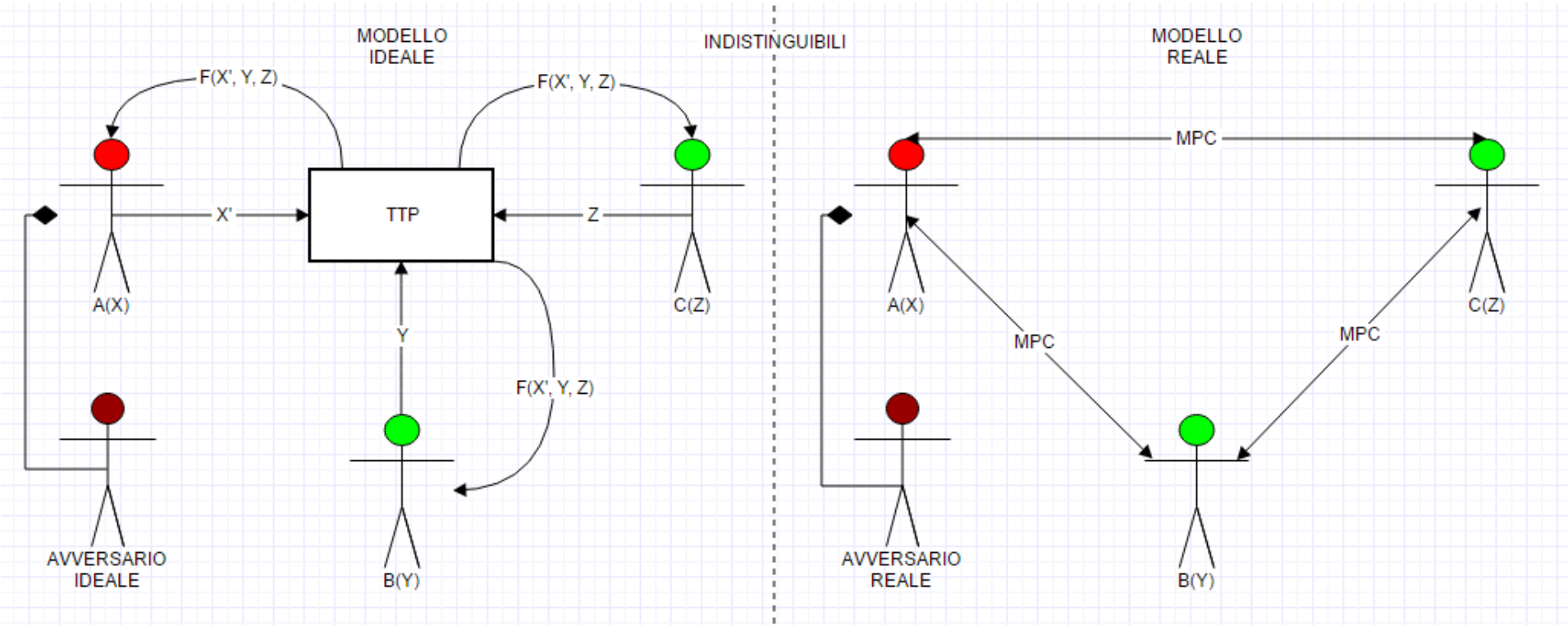
1. (1982) Interesse teorico: protocolli MPC inefficienti.
2. (1998-) Interesse pratico: protocollo efficienti anche se ancora computazionalmente costosi.
3. (2008) Uso pratico: in Danimarca, per la prima volta, un protocollo MPC utilizzato nell'ambito di una doppia asta a livello nazionale.

Possibili usi reali protocolli MPC

- Aste digitali (con offerta segreta).
- Schemi di voto elettronico.

Perché analizzare la MPC usando la Teoria dei Giochi?

- In un *protocollo MPC sicuro* gli unici comportamenti scorretti concessi ad un partecipante sono:
  - Mentire sul proprio valore di input.
  - Non trasmettere alcun valore di input.



Perché analizzare la MPC usando la Teoria dei Giochi?

- partecipanti onesti o disonesti vs partecipanti razionali.
- In un *protocollo MPC sicuro* gli unici comportamenti scorretti concessi ad un partecipante sono:
  - Mentire sul proprio valore di input.
  - Non trasmettere alcun valore di input.

Modellare MPC usando la Teoria dei Giochi

- Come definire un gioco di Multi-Party Computation?
- In quali circostanze partecipanti razionali sono incentivati ad agire correttamente?

Perché analizzare la MPC usando la Teoria dei Giochi?

Modellare MPC usando la Teoria dei Giochi

- Come definire un gioco di Multi-Party Computation?
- In quali circostanze partecipanti razionali sono incentivati ad agire correttamente?

Progettazione di Meccanismi

- (Ω, σ) meccanismo.
- progettare il gioco Ω.
- raccomandare ad ogni P_i di seguire la strategia $\sigma_i \in \sigma$.

Gioco di valutazione di funzione

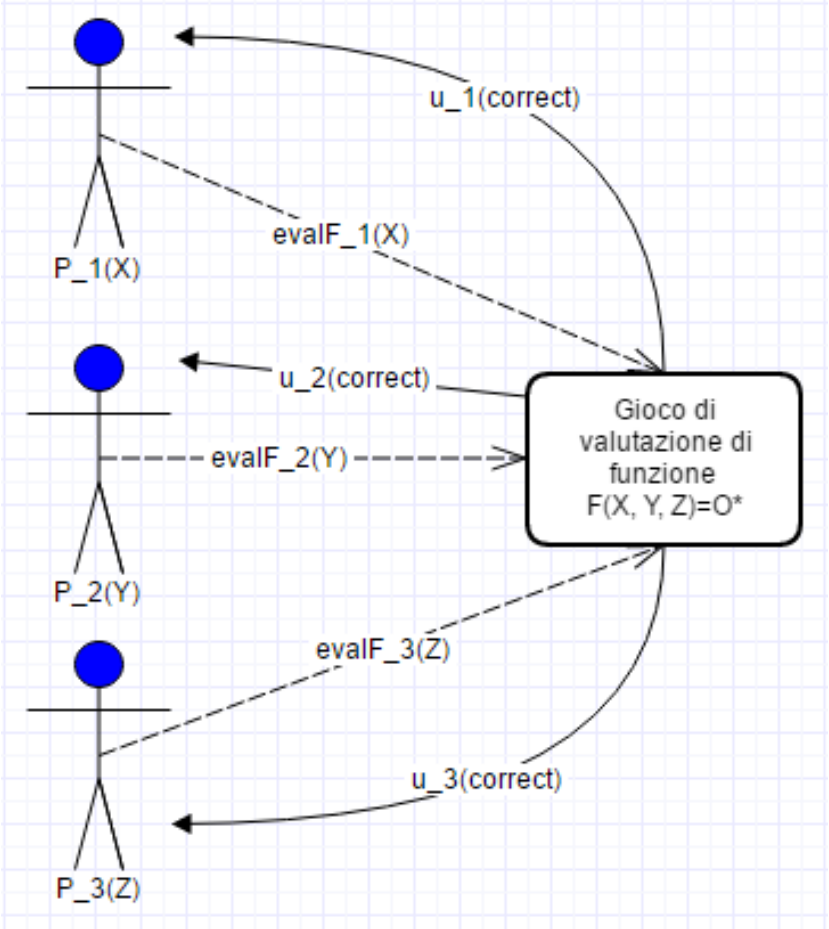
Gioco ad informazione incompleta dove:

- $N = \{P_1, \ldots, P_n\}$, ogni P_i input segreto t_i,
- desiderano stimare correttamente $f(t_1, \ldots, t_n) = o^*$.
- Azioni P_i: stime $eval_i^f(t_i) = o_i$ riguardo all'output o^* di f.
- Utilità P_i: $u_i(correct) = u_i(correct_1, \ldots, correct_n) = u_i(t_1, \ldots, t_n, eval_1^f(t_1), \ldots, eval_n^f(t_n))$.

Come viene giocato?

1. Ogni P_i da una stima dell'output di $f(t)$ (simultaneamente).
2. Ogni P_i riceve il proprio payoff $u_i(correct)$.

Gioco di valutazione di funzione



Considerazioni sulla funzione utilità

- Robert McGrew, Ryan Porter, Yoav Shoham, "Towards a general theory of non-cooperative computation", *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, 2003.

La funzione utilità che descrive le preferenze di ogni P_i dipenderà dalle seguenti considerazioni crittografiche:

- *Correttezza.*
- *Esclusività.*

Vincoli rispettati da $u_i(correct)$ per ogni P_i

- *Vincolo di correttezza.* Ogni volta in cui $correct_i = 1$ e $correct'_i = 0$, $u_i(corret) > u_i(corret')$.
- *Vincolo di esclusività.* Se $corret_i = corret'_i$, per ogni $j \neq i$ abbiamo che $corret_j \leq corret'_j$ fintanto che $corret_j = 0$ e $corret'_j = 1$ per qualche j, allora $u_i(corret) > u_i(corret')$.

Gioco di Multi-Party Computation

- Yevgeniy Dodis, Tal Rabin, "Cryptography and Game Theory", http://www.cs.nyu.edu/~dodis/ps/game-survey.pdf, invited book chapter in *"Algorithmic Game Theory"*, 2007.

- *G gioco di valutazione di funzione f*,
- π protocollo MPC sicuro per la computazione di f,
- G_{MPC} *gioco di Multi-Party Computation* la versione estesa di G tramite cheap-talk costituita dalle seguenti fasi:
  1. *Fase di cheap-talk* (sequenziale): i partecipanti al gioco si scambiano messaggi per un numero finito di round seguendo π in un appropriato *modello di comunicazione*. Conclusa tale fase, ogni P_i possiede $o' = f(t'_1, \ldots, t'_n)$.
  2. *Fase di gioco* (simultanea): ogni P_i partecipa normalmente a G.

Gioco di Multi-Party Computation

- Yevgeniy Dodis, Tal Rabin, "Cryptography and Game Theory", `http://www.cs.nyu.edu/~dodis/ps/game-survey.pdf`, invited book chapter in *"Algorithmic Game Theory"*, 2007.

- *G gioco di valutazione di funzione f*,
- π protocollo MPC sicuro per la computazione di f,
- G_{MPC} *gioco di Multi-Party Computation* la versione estesa di G tramite cheap-talk costituita dalle seguenti fasi:
  1. *Fase di cheap-talk* (sequenziale): π
  2. *Fase di gioco* (simultanea): G

Nota bene

- I payoff dei giocatori in G_{MPC} saranno identificati dai payoff che ottengono in G.
- La strategia s_i di P_i in G_{MPC}: strategia che segue nella fase di esecuzione del protocollo seguita dalla scelta di una stima da giocare in G.

Gioco di Multi-Party Computation

- Yevgeniy Dodis, Tal Rabin, "Cryptography and Game Theory", http://www.cs.nyu.edu/~dodis/ps/game-survey.pdf, invited book chapter in *"Algorithmic Game Theory"*, 2007.

- *G gioco di valutazione di funzione f*,
- π protocollo MPC sicuro per la computazione di f,
- *G_{MPC} gioco di Multi-Party Computation* la versione estesa di G tramite cheap-talk costituita dalle seguenti fasi:
  1. *Fase di cheap-talk* (sequenziale): π
  2. *Fase di gioco* (simultanea): G

Libertà lasciate a P_i:

- (Fase di cheap-talk.) Funzione di condivisione $c_i(t_i) = t_i' \neq t_i$; $c_i(t_i) = \perp$.
- (Fase di gioco.) Stimare f non usando il valore o' ritornato da π.

Gioco di Multi-Party Computation

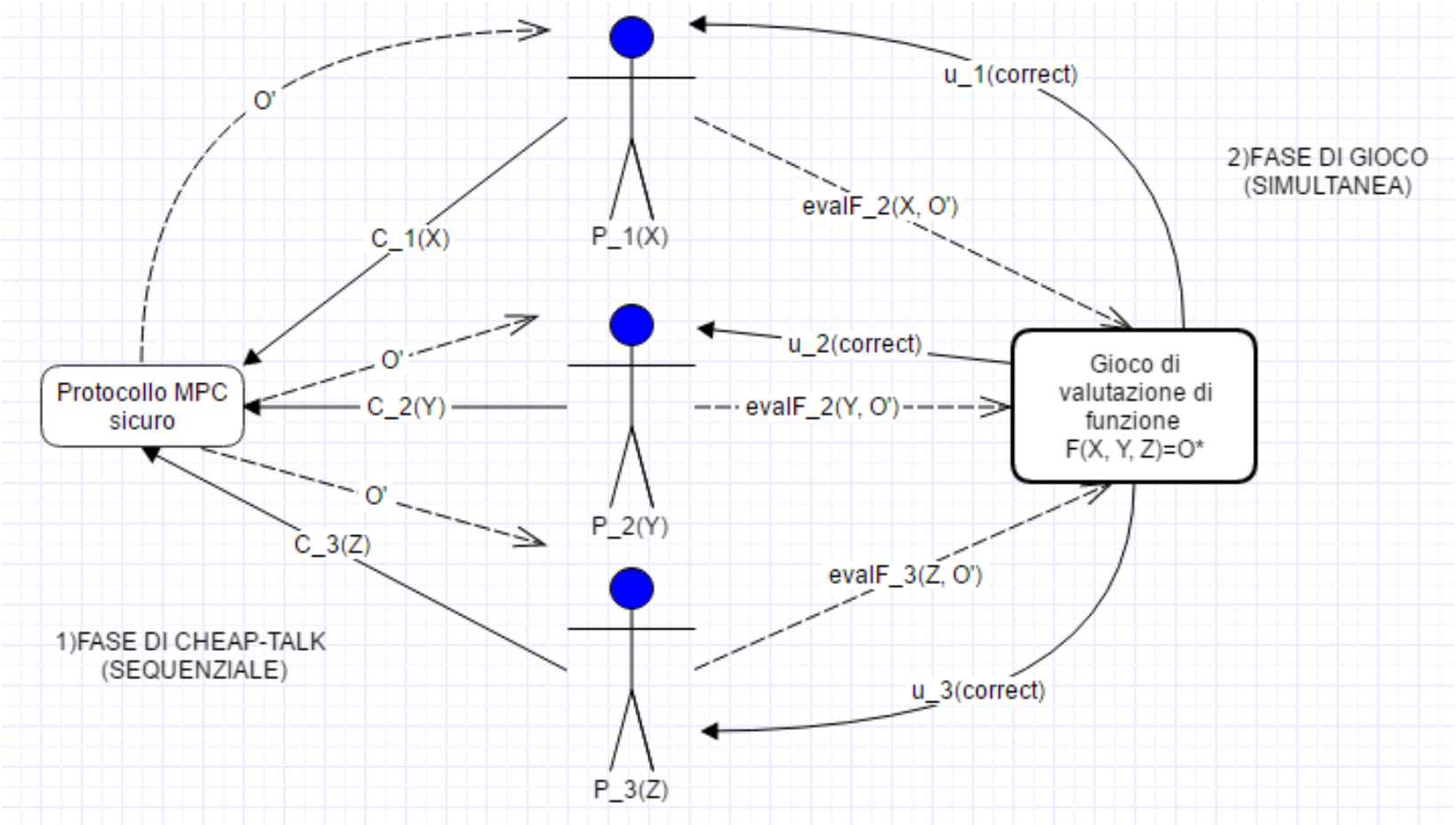
- Yevgeniy Dodis, Tal Rabin, "Cryptography and Game Theory", `http://www.cs.nyu.edu/~dodis/ps/game-survey.pdf`, invited book chapter in *"Algorithmic Game Theory"*, 2007.

- *G gioco di valutazione di funzione f*,
- π protocollo MPC sicuro per la computazione di f,
- G_{MPC} *gioco di Multi-Party Computation* la versione estesa di G tramite cheap-talk costituita dalle seguenti fasi:
  1. *Fase di cheap-talk* (sequenziale): π
  2. *Fase di gioco* (simultanea): G

Domanda:

Quando σ_π, profilo a strategie oneste, risulta equilibrio per G_{MPC}?

Gioco di Multi-Party Computation



Esempio: "Gioco MPC per il calcolo della funzione parità"

Funzione parità $f : \{0,1\}^n \to \{0,1\}$ definita del seguente modo:

$$f(t) = \begin{cases} 1, & \text{se e solo se il numero di 1 nel vettore } t \in \{0,1\}^n \text{ è dispari,} \\ 0, & \text{altrimenti.} \end{cases}$$

In altre parole $f(t) = t_1 \oplus t_2 \oplus \cdots \oplus t_n$.

Strategia di comportamento scorretto

1. P_i condivide $\neg t_i$ durante l'esecuzione di π.
2. P_i stima che il valore corretto di f sarà $\neg o^f$.

Esempio: "Gioco MPC per il calcolo della funzione parità"

Funzione maggioranza $f : \{0,1\}^n \to \{0,1\}$ definita del seguente modo:

$$f(t) = \begin{cases} 1, & \text{se e solo se la maggioranza degli elementi in } t \in \{0,1\}^n \text{ sono 1,} \\ 0, & \text{altrimenti.} \end{cases}$$

In altre parole $f(t) = \lfloor \frac{1}{2} + \frac{(\sum_{i=1}^{n} t_i) - 1/2}{n} \rfloor$.

Comportamento scorretto non incentivato

1. P_i condivide $\neg t_i$ durante l'esecuzione di π.
2. P_i non è in grado di ricostruire il valore corretto della funzione.

Funzioni computabili in modo non-cooperativo (NCC)

- Yoav Shoham, Moshe Tennenholtz, "Non-Cooperative Evaluation of Logical Formulas: The Propositional Case", https://www.researchgate.net/publication/239666328_Non-Cooperative_Evaluation_of_Logical_Formulas_The_Propositional_Case, 2003.
- Yoav Shoham, Moshe Tennenholtz, "Non-cooperative computation: Boolean functions with correctness and exclusivity", *Theoretical Computer Science Volume 343, Issues 1-2*, 2005.

Funzioni computabili in modo non-cooperativo (NCC)

Dato G_{MPC}, diremo che f *è computabile in modo non-cooperativo (NCC)* se vale la seguente proprietà:

- Se P_i mente sul proprio input, non è in grado di ricostruire il corretto valore di f in modo autonomo.
- Se ogni P_i fornisce il proprio input corretto, π suggerisce ad ogni P_i il valore corretto di f.

Funzioni computabili in modo non-cooperativo (NCC)

Teorema Shoham-Tennenholtz

Nel caso in cui: i partecipanti ad un G_{MPC} valutino la proprietà di correttezza maggiormente rispetto alla proprietà di esclusività e $u_{i=1,\ldots,n}(correct)$ rispetta vincoli di correttezza ed esclusività, una funzione è *NCC* se e solo se è non-reversibile e non-dominata.

Funzioni computabili in modo non-cooperativo (NCC)

Teorema Shoham-Tennenholtz

Nel caso in cui: i partecipanti ad un G_{MPC} valutino la proprietà di correttezza maggiormente rispetto alla proprietà di esclusività e $u_{i=1,\ldots,n}(correct)$ rispetta vincoli di correttezza ed esclusività, una funzione è *NCC* se e solo se è non-reversibile e non-dominata.

Classe di funzioni non-NCC: funzioni dominate

Data $f: T^n \rightarrow O$, diremo che f è *dominata* se esiste $t_i \in T$ tale che il valore di f è determinato indipendentemente dagli altri input t_{-i}.
Formalmente: $\forall t_{-i} t'_{-i},\ f(t_i, t_{-i}) = f(t_i, t'_{-i})$

Funzioni computabili in modo non-cooperativo (NCC)

Teorema Shoham-Tennenholtz

Nel caso in cui: i partecipanti ad un G_{MPC} valutino la proprietà di correttezza maggiormente rispetto alla proprietà di esclusività e $u_{i=1,\ldots,n}(correct)$ rispetta vincoli di correttezza ed esclusività, una funzione è *NCC* se e solo se è non-reversibile e non-dominata.

Classe di funzioni non-NCC: funzioni reversibili

Data $f : T^n \rightarrow O$, diremo che f è *reversibile* se, per qualche input $t_i \in T$, esiste un altro input $t'_i \in T$ ed una funzione g tali che:

- (a) $\forall t_{-i}$ abbiamo che $g(f(t'_i, t_{-i}), t_i) = f(t_i, t_{-i})$.
- (b) per qualche t_{-i} abbiamo che $f(t'_i, t_{-i}) \neq f(t_i, t_{-i})$.

Funzioni computabili in modo non-cooperativo (NCC)

Teorema Shoham-Tennenholtz

Nel caso in cui: i partecipanti ad un G_{MPC} valutino la proprietà di correttezza maggiormente rispetto alla proprietà di esclusività e $u_{i=1,\ldots,n}(correct)$ rispetta vincoli di correttezza ed esclusività, una funzione è *NCC* se e solo se è non-reversibile e non-dominata.

Corollario

Ogni funzione che è reversibile o dominata è *non-NCC*.

Giochi MPC con coalizioni

- Itai Ashlagi, Andrey Klinger, Moshe Tenneholtz, "K-NCC: Stability Against Group Deviations in Non-cooperative Computation", *Internet and Network Economics, Volume 4858 of the series Lecture Notes in Computer Science*, 2007.

Generalizzazione funzioni NCC in caso di coalizioni

funzioni K-NCC.

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC (K-NCC $\forall 1 \leq k \leq n$) se e solo se è NCC.

Giochi MPC con coalizioni

- Ittai Abraham, Danny Dolev, Rica Gonen, Joe Halpern, "Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation", *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, 2006.

Generalizzazione equilibrio di Nash per coalizioni

equilibrio di Nash K-resistente.

Risultato Dodis-Rabin

- Yevgeniy Dodis, Tal Rabin, "Cryptography and Game Theory", `http://www.cs.nyu.edu/~dodis/ps/game-survey.pdf`, invited book chapter in *"Algorithmic Game Theory"*, 2007.

Corollario Dodis-Rabin

- Se π protocollo MPC che computa f in modo sicuro contro coalizioni di al massimo K partecipanti.
- e f è K-NCC,
- allora σ_π è equilibrio K-resistente in G_{MPC} in cui si esegue π per il calcolo di f.

Conclusioni

1. (G_{MPC}, σ_π) *meccanismo K-resistente per la valutazione di funzioni K-NCC*
   - Le parti eseguono π per il calcolo di f non solo da un punto di vista crittografico, ma anche dal punto di vista razionale della Teoria dei Giochi.
   - Sappiamo che i giocatori saranno incentivati ad agire in modo corretto.
2. Nel campo economico della progettazione di meccanismi molti *meccanismi pratici* richiedono un mediatore: implementarlo utilizzando un protocollo di Multi-Party Computation.

Conclusioni

1. (G_{MPC}, σ_π) *meccanismo K-resistente per la valutazione di funzioni K-NCC*
   - Le parti eseguono π per il calcolo di f non solo da un punto di vista crittografico, ma anche dal punto di vista razionale della Teoria dei Giochi.
   - Sappiamo che i giocatori saranno incentivati ad agire in modo corretto.
2. Nel campo economico della progettazione di meccanismi molti *meccanismi pratici* richiedono un mediatore: implementarlo utilizzando un protocollo di Multi-Party Computation.
   - Yevgeniy Dodis, Shai Halevi, Tal Rabin, "A Cryptographic Solution to a Game Theoretic Problem", *Advances in Cryptology*, 2000.

Condizione sufficiente

- in G_M il profilo a strategie oneste deve essere equilibrio correlato
- il protocollo MPC in questione deve *computare in modo sicuro* la funzione probabilistica che descrive la strategia di selezione delle azioni raccomandate dal mediatore.

Problemi aperti e sviluppi futuri

1. Problema aperto: estendere tali risultati anche per funzioni non-NCC:
   - classe limitata di funzioni;
   - stabilire se una funzione è K-NCC è un problema computazionalmente difficile.

Risultati attuali per le funzioni K-NCC

Qualunque funzione booleana simmetrica ad n variabili è fortemente-NCC (K-NCC per ogni $K = k$ con $1 \leq k < n$) se e solo se è NCC.

- ad esempio: funzione maggioranza è fortemente-NCC.

2. Problema: l'equilibrio di Nash nei giochi in forma estesa prende in considerazione anche *storie irrealizzabili*
   - Sviluppi futuri: progettazione di protocolli MPC sufficientemente robusti per raggiungere situazioni di equilibrio più "forti".

Problemi aperti e sviluppi futuri

1. Problema aperto: estendere tali risultati anche per funzioni non-NCC:
   - classe limitata di funzioni;
   - stabilire se una funzione è K-NCC è un problema computazionalmente difficile.
2. Problema: l'equilibrio di Nash nei giochi in forma estesa prende in considerazione anche *storie irrealizzabili*
   - Sviluppi futuri: progettazione di protocolli MPC sufficientemente robusti per raggiungere situazioni di equilibrio più "forti".

Bibliografia parziale

- John von Neumann, Oskar Morgenstern, "Theory of Games and Economic Behavior", *Princeton University Press*, 1944.
- Andrew C. Yao, "Protocols for secure computations", *FOCS. 23rd Annual Symposium on Foundations of Computer Science*, 1982.
- Yevgeniy Dodis, Shai Halevi, Tal Rabin, "A Cryptographic Solution to a Game Theoretic Problem", *Advances in Cryptology*, 2000.
- Yoav Shoham, Moshe Tennenholtz, "Non-Cooperative Evaluation of Logical Formulas: The Propositional Case", `https://www.researchgate.net/publication/239666328_Non-Cooperative_Evaluation_of_Logical_Formulas_The_Propositional_Case`, 2003.

Bibliografia parziale

- Robert McGrew, Ryan Porter, Yoav Shoham, "Towards a general theory of non-cooperative computation", *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, 2003.
- Yoav Shoham, Moshe Tennenholtz, "Non-cooperative computation: Boolean functions with correctness and exclusivity", *Theoretical Computer Science Volume 343, Issues 1-2*, 2005.
- Ittai Abraham, Danny Dolev, Rica Gonen, Joe Halpern, "Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation", *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, 2006.
- Itai Ashlagi, Andrey Klinger, Moshe Tenneholtz, "K-NCC: Stability Against Group Deviations in Non-cooperative Computation", *Internet and Network Economics, Volume 4858 of the series Lecture Notes in Computer Science*, 2007.

Bibliografia parziale

- Yevgeniy Dodis, Tal Rabin, "Cryptography and Game Theory", `http://www.cs.nyu.edu/~dodis/ps/game-survey.pdf`, invited book chapter in *"Algorithmic Game Theory"*, 2007.
- Katz J., "Bridging game theory and cryptography: recent results and future directions", *Proceedings of the 5th conference on Theory of cryptography*, 2008.
- Peter Bogetoft, Dan Lund Christensen, Ivan Damgard, Martin Geisler, Thomas Jakobsen, Mikkel Krigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, Tomas Tofty, "Secure Multiparty Computation Goes Live", *Financial Cryptography and Data Security: 13th International Conference*, 2009.

Grazie per l'attenzione.

Strategia nei Giochi in forma normale

Sia $G = (N, A, u)$ un gioco in forma normale:

- L'*insieme di strategie miste* per il giocatore$_i$ sarà $S_i = \Delta(A_i)$, cioè l'insieme di tutte le distribuzioni di probabilità su A_i.
- Una *strategia mista* per il giocatore-i sarà $s_k(A_i) = \{s_k(a_1), \ldots, s_k(a_m)\}$ un vettore di probabilità sulle possibili m azioni che il giocatore-i può intraprendere ($s_k(a_j)$ la probabilità che l'azione $a_j \in A_i$ venga giocata nella particolare strategia mista s_k).
- Il vettore $s = (s_1, \ldots, s_n)$ è un *profilo a strategie miste*, $s_i \in S_i$.

Funzione utilità attesa

Dato un gioco in forma normale $G = (N, A, u)$ ed un profilo a strategia mista $s = (s_1, \ldots, s_n)$, la *funzione utilità attesa* per il giocatore$-i$ sarà

$$u_i(s) = \sum_{a \in A} u_i(a) \cdot \prod_{j=1}^{n} s_j(a_j)$$

con $a \in A$ profilo di azioni.

Possibili usi reali protocolli MPC

- Aste digitali (con offerta segreta).
- Schemi di voto elettronico.

Funzione asta digitale (con offerta segreta)

$f(x_1, \ldots, x_n) = (x_j, j)$,
dove $1 \leq j \leq n$ è l'indice per cui $x_j = max(x_1, \ldots, x_n)$.

Funzione maggioranza binaria

$$f(x_1, \ldots, x_n) = \begin{cases} 1, & \text{se } \sum_{i=1}^{n} x_i > \frac{n}{2}, \\ 0, & \text{altrimenti.} \end{cases}$$

Avversari nella Multi-Party Computation

- Un protocollo MPC deve permettere di calcolare il valore di una data funzione anche sotto l'assunzione che qualcuno dei partecipanti possa deviare dal protocollo stesso : partecipanti onesti oppure disonesti.
- Avversario $\mathcal{A}$ controlla $k < n$ (valore di soglia) partecipanti.

Comportamento degli avversari

- Avversario onesto-ma-curioso (corruzione passiva).
- Avversario malevolo (corruzione attiva).

Potere computazionale degli avversari

- Avversario limitato computazionalmente.
- Avversario illimitato computazionalmente.

Sicurezza nei protocolli MPC: requisiti di sicurezza MPC

Requisiti di sicurezza MPC

- *Privacy*.
- *Correttezza della computazione*.
- *Equità (Fairness)*.
- *Consegna dell'output garantita*.

- Nel *modello ideale* esiste T terza parte fidata, dunque i *requisiti di sicurezza MPC* sono garantiti.
- Nel *modello reale* non esiste alcun T, i partecipanti alla computazione eseguono un protocollo MPC π; quando possiamo ritenere un protocollo MPC sicuro?
  - La sicurezza di un protocollo MPC è stabilita comparando i risultati dell'esecuzione del protocollo reale con i risultati di una computazione ideale nel modello ideale.

Sicurezza nei protocolli MPC: computazione sicura

- Computazione sicura:
  - $\mathcal{A}$ ($\mathcal{A}'$): avversario che controlla $k < n$ partecipanti nel *modello reale* (*modello ideale*).
  - t: vettore degli input dei partecipanti alla computazione.
  - λ: parametro di sicurezza.
  - $REAL_{\mathcal{A},\pi}(t,\lambda)$: trascrizione reale (output di $\mathcal{A}$ unito all'output dei partecipanti onesti dopo aver eseguito π con ingresso t).
  - $IDEAL_{\mathcal{A}',f}(t,\lambda)$: trascrizione ideale (output di $\mathcal{A}'$ unito all'output dei partecipanti calcolati da T con input il vettore t).
  - *π calcola in modo sicuro f* se per ogni $\mathcal{A}$ esiste $\mathcal{A}'$ tale che gli insieme di distribuzioni $REAL_{\mathcal{A},\pi} = \{REAL_{\mathcal{A},\pi}(t,\lambda)\}_{t\in\{0,1\}^*,\lambda\in\mathbb{N}}$ e $IDEAL_{\mathcal{A}',f} = \{IDEAL_{\mathcal{A}',f}(t,\lambda)\}_{t\in\{0,1\}^*,\lambda\in\mathbb{N}}$ sono tra loro "indistinguibili".

Sicurezza nei protocolli MPC: garanzie di sicurezza nei protocollo MPC sicuri

- In un protocollo π che *calcola in modo sicuro* f (*protocollo MPC sicuro*) i *requisiti di sicurezza MPC* sono garantiti.

Comportamenti scorretti ammessi in un protocollo MPC sicuro

Un partecipante corrotto potrà:

- Mentire sul proprio valore di input.
- Non trasmettere alcun valore di input.

Modello di teoria dell'informazione

- Avversario illimitato computazionalmente.
- Comunicazione tramite canali sicuri.
- Sicurezza perfetta:
  - $REAL_{\mathcal{A},\pi} \stackrel{d}{\equiv} IDEAL_{\mathcal{A}',f}$: i due insiemi di distribuzioni sono *identicamente distribuiti* se, per ogni t e per ogni λ le distribuzioni $REAL_{\mathcal{A},\pi}(t,\lambda)$ e $IDEAL_{\mathcal{A}',f}(t,\lambda)$ sono identiche.
- Sicurezza statistica:
  - Distanza statistica: $\Delta(X_0, X_1) = \frac{1}{2} \cdot \sum_{x \in \mathcal{X}} |\mathbb{P}[X_0 = x] - \mathbb{P}[X_1 = x]|$, con X_0 e X_1 variabili aleatorie con valori in $\mathcal{X}$ insieme numerabile.
  - $REAL_{\mathcal{A},\pi} \stackrel{s}{\approx} IDEAL_{\mathcal{A}',f}$: i due insiemi di distribuzioni sono *statisticamente indistinguibili* se, per tutti i t e per tutti i λ sufficientemente grandi abbiamo che $\Delta(REAL_{\mathcal{A},\pi}(t,\lambda), IDEAL_{\mathcal{A}',f}(t,\lambda)) < \sigma(\lambda)$, con σ funzione trascurabile.

Modello di teoria dell'informazione

- Avversario illimitato computazionalmente.
- Comunicazione tramite canali sicuri.
- Sicurezza perfetta:
  - $REAL_{\mathcal{A},\pi} \stackrel{d}{\equiv} IDEAL_{\mathcal{A}',f}$: i due insiemi di distribuzioni sono *identicamente distribuiti* se, per ogni t e per ogni λ le distribuzioni $REAL_{\mathcal{A},\pi}(t,\lambda)$ e $IDEAL_{\mathcal{A}',f}(t,\lambda)$ sono identiche.
- Sicurezza statistica:
  - Distanza statistica: $\Delta(X_0, X_1) = \frac{1}{2} \cdot \sum_{x \in \mathcal{X}} |\mathbb{P}[X_0 = x] - \mathbb{P}[X_1 = x]|$, con X_0 e X_1 variabili aleatorie con valori in $\mathcal{X}$ insieme numerabile.
  - $REAL_{\mathcal{A},\pi} \stackrel{s}{\approx} IDEAL_{\mathcal{A}',f}$: i due insiemi di distribuzioni sono *statisticamente indistinguibili* se, per tutti i t e per tutti i λ sufficientemente grandi abbiamo che $\Delta(REAL_{\mathcal{A},\pi}(t,\lambda), IDEAL_{\mathcal{A}',f}(t,\lambda)) < \sigma(\lambda)$, con σ funzione trascurabile.

Modello computazionale

- Avversario e partecipanti limitati computazionalmente.
  - Sia λ il parametro di sicurezza:
    - Tutti i calcoli e le computazioni sono realizzati in un numero di passi n polinomiale in λ
    - Le strategie di comportamento scorretto dei partecipanti corrotti sono limitate ad essere eseguite in tempo polinomiale in λ.
- Comunicazione tramite canali autenticati.
- Sicurezza computazionale:
  - $REAL_{\mathcal{A},\pi} \overset{c}{\approx} IDEAL_{\mathcal{A}',f}$: i due insiemi di distribuzioni sono *computazionalmente indistinguibili* se, per ogni attaccante $\mathcal{A}$ ($\mathcal{A}'$), per tutti i t e per tutti i λ sufficientemente grandi abbiamo che $|\mathbb{P}[\mathcal{A}(REAL_{\mathcal{A},\pi}(t,\lambda),1^{\lambda})=1] - \mathbb{P}[\mathcal{A}'(IDEAL_{\mathcal{A}',f}(t,\lambda),1^{\lambda})=1]| < \sigma(\lambda)$, con σ funzione trascurabile.

Risultati classici per MPC sicura nel caso di avversari a soglia

Modello	Avversario	Condizione
computazionale	passivo	$k < n$
computazionale	attivo	$k < \frac{n}{2}$
teoria dell'informazione	passivo	$k < \frac{n}{2}$
teoria dell'informazione	attivo	$k < \frac{n}{3}$
teoria dell'informazione (con canale di broadcast)	attivo	$k < \frac{n}{2}$

Tabella: *Condizioni di soglia necessarie e sufficienti affinché MPC sicura sia possibile.*

Condivisione di segreti di Shamir di tipo-(n, k)

Il protocollo riceve in input un segreto $s \in S = \mathbb{Z}_p$, per qualche primo $p > n > k$

- *Condivisione del segreto.*
  1. Casualmente si scelgono i coefficienti $a_1, \ldots, a_k$ da $\mathbb{Z}_p$.
  2. Si pone $f(X) = s + a_1 \cdot X + a_2 \cdot X^2 + \cdots + a_k \cdot X^k$.
  3. Per ogni $i = 1, 2, \ldots, n$, calcoliamo le porzioni di segreto $s_i = f(i) \bmod p$.
- *Ricostruzione del segreto.* In tale fase le porzioni $\{s_i\}_{i \in \Phi}$ vengono utilizzate per ricostruire il segreto, Φ un qualsiasi sottoinsieme di $\{1, \ldots, n\}$ con $k + 1$ elementi.
  1. $\alpha_i = \prod_{\substack{j \in \Phi \\ j \neq i}} \frac{j}{j - i}$.
  2. $s = \sum_{i \in \Phi} \alpha_i \cdot s_i$
- **Nota bene**: $k + 1$ porzioni sono sufficienti per ottenere $g(X)$ il polinomio interpolante della funzione $f(X)$ e sarà perciò possibile calcolare $g(0) = s = f(0)$. $g(X) = \sum_{i \in \Phi} s_i \cdot \prod_{\substack{j \in \Phi \\ j \neq i}} \frac{j - X}{j - i}$.

Condivisione di segreti di Shamir di tipo-(n, k)

Il protocollo riceve in input un segreto $s \in S = \mathbb{Z}_p$, per qualche primo $p > n > k$

- *Condivisione del segreto.*
  1. Casualmente si scelgono i coefficienti $a_1, \ldots, a_k$ da $\mathbb{Z}_p$.
  2. Si pone $f(X) = s + a_1 \cdot X + a_2 \cdot X^2 + \cdots + a_k \cdot X^k$.
  3. Per ogni $i = 1, 2, \ldots, n$, calcoliamo le porzioni di segreto $s_i = f(i) \bmod p$.

- *Ricostruzione del segreto.* In tale fase le porzioni $\{s_i\}_{i \in \Phi}$ vengono utilizzate per ricostruire il segreto, Φ un qualsiasi sottoinsieme di $\{1, \ldots, n\}$ con $k + 1$ elementi.
  1. $\alpha_i = \prod_{\substack{j \in \Phi \\ j \neq i}} \frac{j}{j-i}$.
  2. $s = \sum_{i \in \Phi} \alpha_i \cdot s_i$
- **Nota bene**: $k + 1$ porzioni sono sufficienti per ottenere $g(X)$ il polinomio interpolante della funzione $f(X)$ e sarà perciò possibile calcolare $g(0) = s = f(0)$. $g(X) = \sum_{i \in \Phi} s_i \cdot \prod_{\substack{j \in \Phi \\ j \neq i}} \frac{j-X}{j-i}$.

Condivisione di segreti di Shamir di tipo-(n, k)

Il protocollo riceve in input un segreto $s \in S = \mathbb{Z}_p$, per qualche primo $p > n > k$

- *Condivisione del segreto.*
  1. Casualmente si scelgono i coefficienti $a_1, \ldots, a_k$ da $\mathbb{Z}_p$.
  2. Si pone $f(X) = s + a_1 \cdot X + a_2 \cdot X^2 + \cdots + a_k \cdot X^k$.
  3. Per ogni $i = 1, 2, \ldots, n$, calcoliamo le porzioni di segreto $s_i = f(i) \bmod p$.
- *Ricostruzione del segreto.* In tale fase le porzioni $\{s_i\}_{i \in \Phi}$ vengono utilizzate per ricostruire il segreto, Φ un qualsiasi sottoinsieme di $\{1, ..., n\}$ con $k + 1$ elementi.
  1. $\alpha_i = \prod_{\substack{j \in \Phi \\ j \neq i}} \frac{j}{j-i}$.
  2. $s = \sum_{i \in \Phi} \alpha_i \cdot s_i$
- **Nota bene**: $k + 1$ porzioni sono sufficienti per ottenere $g(X)$ il polinomio interpolante della funzione $f(X)$ e sarà perciò possibile calcolare $g(0) = s = f(0)$. $g(X) = \sum_{i \in \Phi} s_i \cdot \prod_{\substack{j \in \Phi \\ j \neq i}} \frac{j-X}{j-i}$.

Esempio: "Problema MPC semplice"

- Input segreti $x_{i=1,\ldots,n} \in \mathbb{F}$.
- Comunicazione su *canali sicuri*.
- Avversario a soglia *onesto-ma-curioso*.
- Vogliamo garantire *input privacy* e *correttezza della computazione*.
- La funzione da computare calcolabile attraverso un *circuito aritmetico* che gestisce elementi del campo $\mathbb{F}$.

Circuito aritmetico

Un *circuito aritmetico* con n input è un grafo senza cicli in cui:

1. esiste un unico nodo con grado di uscita 0, l'output del circuito.
2. esistono n nodi con grado di ingresso 1, gli input del circuito.
3. ogni nodo interno ha grado di ingresso 2 e grado di uscita 1 ed è un nodo di moltiplicazione oppure un nodo di addizione.

Esempio: "Problema MPC semplice"

- Input segreti $x_{i=1,\ldots,n} \in \mathbb{F}$.
- Comunicazione su *canali sicuri*.
- Avversario a soglia *onesto-ma-curioso*.
- Vogliamo garantire *input privacy* e *correttezza della computazione*.
- La funzione da computare calcolabile attraverso un *circuito aritmetico* che gestisce elementi del campo $\mathbb{F}$.

Circuito aritmetico

Un *circuito aritmetico* con n input è un grafo senza cicli in cui:

1. esiste un unico nodo con grado di uscita 0, l'output del circuito.
2. esistono n nodi con grado di ingresso 1, gli input del circuito.
3. ogni nodo interno ha grado di ingresso 2 e grado di uscita 1 ed è un nodo di moltiplicazione oppure un nodo di addizione.

Protocollo MPC semplice

Ogni giocatore P_i, con $i \in \{1, \ldots, n\}$, possiede un input segreto $x_i \in \mathbb{F}$.

- *Condivisione degli input.* P_i condivide x_i usando uno schema di Shamir di tipo-(n, k). Siano $s_1^{(i)}, \ldots, s_n^{(i)}$ le porzioni risultanti. Ogni giocatore P_i invia $s_j^{(i)}$ al giocatore P_j.

Protocollo MPC semplice

- *Computazione.* Per $c = n+1, \ldots, d$, si calcola l'output del nodo corrente G_c nel seguente modo:
  - Supponiamo che i lati in ingresso al nodo G_c provengano dai nodi G_i, G_j per $i, j < c$ ed indichiamo con $s_1^{(i)}, \ldots, s_n^{(i)}$ ed $s_1^{(j)}, \ldots, s_n^{(j)}$ le porzioni dell'output dei nodi G_i e G_j.
  - Se G_c è un nodo di addizione: ogni giocatore $P_{r=\{1,\ldots,n\}}$ calcola localmente $s_r^{(c)} = s_r^{(i)} + s_r^{(j)}$, la porzione della somma degli output dei nodi G_i e G_j.
  - Se G_c è un nodo di moltiplicazione:
    1. Ogni giocatore $P_{r=\{1,\ldots,n\}}$ calcola localmente $s_r^{(2k,c)} = s_r^{(i)} \cdot s_r^{(j)}$, porzione relativa al prodotto degli output dei nodi G_i e G_j in uno schema di Shamir di tipo-$(n, 2k)$.
    2. P_r genera le porzioni relative ad $s_r^{(2k,c)}$ con uno schema di Shamir di tipo-(n, k): $s_1^{(r)}, \ldots, s_n^{(r)}$. Ogni giocatore $P_{k=\{1,\ldots,n\}}$ invia $s_r^{(k)}$ a P_r.
    3. Ogni P_r usando la fase di ricostruzione di uno schema di Shamir di tipo-$(n, 2k)$ calcola $s_r^{(c)}$, la porzione del prodotto degli output dei nodi G_i e G_j in uno schema di Shamir di tipo-(n, k).

Protocollo MPC semplice

- *Computazione.* Per $c = n+1, \ldots, d$, si calcola l'output del nodo corrente G_c nel seguente modo:
  - Supponiamo che i lati in ingresso al nodo G_c provengano dai nodi G_i, G_j per $i, j < c$ ed indichiamo con $s_1^{(i)}, \ldots, s_n^{(i)}$ ed $s_1^{(j)}, \ldots, s_n^{(j)}$ le porzioni dell'output dei nodi G_i e G_j.
  - Se G_c è un nodo di addizione: ogni giocatore $P_{r=\{1,\ldots,n\}}$ calcola localmente $s_r^{(c)} = s_r^{(i)} + s_r^{(j)}$, la porzione della somma degli output dei nodi G_i e G_j.
  - Se G_c è un nodo di moltiplicazione:
    1. Ogni giocatore $P_{r=\{1,\ldots,n\}}$ calcola localmente $s_r^{(2k,c)} = s_r^{(i)} \cdot s_r^{(j)}$, porzione relativa al prodotto degli output dei nodi G_i e G_j in uno schema di Shamir di tipo-$(n, 2k)$.
    2. P_r genera le porzioni relative ad $s_r^{(2k,c)}$ con uno schema di Shamir di tipo-(n, k): $\tilde{s}_1^{(r)}, \ldots, \tilde{s}_n^{(r)}$. Ogni giocatore $P_{k=\{1,\ldots,n\}}$ invia $\tilde{s}_r^{(k)}$ a P_r.
    3. Ogni P_r usando la fase di ricostruzione di uno schema di Shamir di tipo-$(n, 2k)$ calcola $s_r^{(c)}$, la porzione del prodotto degli output dei nodi G_i e G_j in uno schema di Shamir di tipo-(n, k).

Protocollo MPC semplice

- *Computazione.* Per $c = n+1, \ldots, d$, si calcola l'output del nodo corrente G_c nel seguente modo:
  - Supponiamo che i lati in ingresso al nodo G_c provengano dai nodi G_i, G_j per $i, j < c$ ed indichiamo con $s_1^{(i)}, \ldots, s_n^{(i)}$ ed $s_1^{(j)}, \ldots, s_n^{(j)}$ le porzioni dell'output dei nodi G_i e G_j.
  - Se G_c è un nodo di addizione: ogni giocatore $P_{r=\{1,\ldots,n\}}$ calcola localmente $s_r^{(c)} = s_r^{(i)} + s_r^{(j)}$, la porzione della somma degli output dei nodi G_i e G_j.
  - Se G_c è un nodo di moltiplicazione:
    1. Ogni giocatore $P_{r=\{1,\ldots,n\}}$ calcola localmente $s_r^{(2k,c)} = s_r^{(i)} \cdot s_r^{(j)}$, porzione relativa al prodotto degli output dei nodi G_i e G_j in uno schema di Shamir di tipo-$(n, 2k)$.
    2. P_r genera le porzioni relative ad $s_r^{(2k,c)}$ con uno schema di Shamir di tipo-(n, k): $\bar{s}_1^{(r)}, \ldots, \bar{s}_n^{(r)}$. Ogni giocatore $P_{k=\{1,\ldots,n\}}$ invia $\bar{s}_r^{(k)}$ a P_r.
    3. Ogni P_r usando la fase di ricostruzione di uno schema di Shamir di tipo-$(n, 2k)$ calcola $s_r^{(c)}$, la porzione del prodotto degli output dei nodi G_i e G_j in uno schema di Shamir di tipo-(n, k).

Protocollo MPC semplice

- *Rivelazione dell'output.* Ogni giocatore P_i invia la porzione $s_i^{(d)}$ relativa al nodo G_d, nodo output del circuito, al giocatore P_1. P_1 eseguendo la fase di ricostruzione dello schema di Shamir di tipo-(n, k) ricostruisce l'output del circuito $s = f(x_1, \ldots, x_n)$ a partire da $k + 1$ porzioni ed invia il valore ottenuto a tutti gli altri giocatori.

Gioco ad informazione incompleta

Gioco ad informazione incompleta (N, A, T, p, u), dove:

- N è l'insieme degli n giocatori; $P_i \in N$.
- $A = A_1 \times \cdots \times A_n$, dove A_i è l'insieme delle possibili azioni di P_i.
- $T = T_1 \times \cdots \times T_n$, dove T_i è lo spazio dei tipi del giocatore P_i.
- $p\colon T \to [0, 1]$ è una distribuzione di probabilità sui vettori di tipi $(t_1, \ldots, t_n)$.
- $u = (u_1, \ldots, u_n)$, dove $u_i\colon A \times T \to \mathbb{R}$ è la funzione utilità per P_i.

1. $(t_1, \ldots, t_n)$ è scelto in accordo con p e t_i assegnato a P_i.
2. Ogni P_i giocherà un'azione $a_i \in A_i$ (simultaneamente).
3. Ogni P_i riceve il proprio payoff $u_i(t_1, \ldots, t_n, a_1, \ldots, a_n)$.

Gioco ad informazione incompleta

Gioco ad informazione incompleta (N, A, T, p, u), dove:

- N è l'insieme degli n giocatori; $P_i \in N$.
- $A = A_1 \times \cdots \times A_n$, dove A_i è l'insieme delle possibili azioni di P_i.
- $T = T_1 \times \cdots \times T_n$, dove T_i è lo spazio dei tipi del giocatore P_i.
- $p\colon T \to [0, 1]$ è una distribuzione di probabilità sui vettori di tipi $(t_1, \ldots, t_n)$.
- $u = (u_1, \ldots, u_n)$, dove $u_i\colon A \times T \to \mathbb{R}$ è la funzione utilità per P_i.

1. $(t_1, \ldots, t_n)$ è scelto in accordo con p e t_i assegnato a P_i.
2. Ogni P_i giocherà un'azione $a_i \in A_i$ (simultaneamente).
3. Ogni P_i riceve il proprio payoff $u_i(t_1, \ldots, t_n, a_1, \ldots, a_n)$.

Equilibrio correlato

- *Equilibri di Nash* non interessanti in un gioco di valutazione di funzione.

Equilibrio correlato

Dato un *gioco in forma normale mediato* $G = (N, A, u)$, una distribuzione $\mathcal{M} \in \Delta(A)$ è un *equilibrio correlato* se, per ogni P_i e per una qualunque funzione di deviazione $r_i : A_i \to A_i$, vale che $u_i(\mathcal{M}) \geq u_i(r_i(a_i^*), a_{-i}^*)$ (dove $a^* = (a_1^*, \ldots, a_n^*) = (a_i^*, a_{-i}^*)$ è un campione di azioni selezionato in accordo con $\mathcal{M}$).

Gioco mediato di valutazione di funzione

Dato un mediatore M in grado di calcolare il valore di una funzione, G_M versione mediata di G *gioco di valutazione di funzione* è un gioco che consiste di quattro fasi:

1. Input privati $(t_1, \ldots, t_n)$ scelti in accordo con p. Ad ogni P_i viene assegnato t_i.
2. Ogni P_i invia un tipo/input t'_i al mediatore M.
3. M calcola il $o' = f(t'_1, \ldots, t'_n)$ e raccomanda ad ogni P_i di giocare o'.
4. Ogni giocatore partecipa normalmente a G.

Gioco mediato di valutazione di funzione

Dato un mediatore M in grado di calcolare il valore di una funzione, G_M versione mediata di *G gioco di valutazione di funzione* è un gioco che consiste di quattro fasi:

1. Input privati $(t_1, \ldots, t_n)$ scelti in accordo con p. Ad ogni P_i viene assegnato t_i.
2. Ogni P_i invia un tipo/input t'_i al mediatore M.
3. M calcola il $o' = f(t'_1, \ldots, t'_n)$ e raccomanda ad ogni P_i di giocare o'.
4. Ogni giocatore partecipa normalmente a G.

Libertà lasciate a P_i:

- Inviare un tipo sbagliato $t'_i \neq t_i$, oppure non inviare alcun tipo, ad M.
- Decidere di non seguire la raccomandazione e giocare un azione $o_i \neq o'_i$.

Gioco mediato di valutazione di funzione

Dato un mediatore M in grado di calcolare il valore di una funzione, G_M versione mediata di *G gioco di valutazione di funzione* è un gioco che consiste di quattro fasi:

1. Input privati $(t_1, \ldots, t_n)$ scelti in accordo con p. Ad ogni P_i viene assegnato t_i.
2. Ogni P_i invia un tipo/input t'_i al mediatore M.
3. M calcola il $o' = f(t'_1, \ldots, t'_n)$ e raccomanda ad ogni P_i di giocare o'.
4. Ogni giocatore partecipa normalmente a G.

Strategia onesta

- P_i fornisce il proprio input corretto $t'_i = t_i$ ad M.
- P_i stima la funzione seguendo il suggerimento di M.

Gioco mediato di valutazione di funzione

Dato un mediatore M in grado di calcolare il valore di una funzione, G_M versione mediata di *G gioco di valutazione di funzione* è un gioco che consiste di quattro fasi:

1. Input privati $(t_1, \ldots, t_n)$ scelti in accordo con p. Ad ogni P_i viene assegnato t_i.
2. Ogni P_i invia un tipo/input t'_i al mediatore M.
3. M calcola il $o' = f(t'_1, \ldots, t'_n)$ e raccomanda ad ogni P_i di giocare o'.
4. Ogni giocatore partecipa normalmente a G.

Nota bene

- *Profilo a strategie oneste* permette ad ogni P_i di stimare correttamente f.
- Per quale classi di funzioni il *profilo a strategie oneste* forma un equilibrio correlato?

Funzioni K-NCC

Dato un *gioco mediato di valutazione di funzione* f, diremo che f è *K-NCC* se per ogni sottoinsieme di k giocatori che formano una coalizione $C_i = \{P_{i_1}, \ldots, P_{i_k}\}$ con $k \leq K$, vale la seguente proprietà:

- Se $P_{i_j} \in C_i$ mente sul proprio input, la coalizione C_i non è in grado di ricostruire il corretto valore di f in modo autonomo.
- Se ogni P_i fornisce il proprio input corretto ad M, ad ogni P_i viene suggerito il valore corretto di f.

Funzioni K-NCC

Dato un *gioco mediato di valutazione di funzione* f, diremo che f è *K-NCC* se per ogni sottoinsieme di k giocatori che formano una coalizione $C_i = \{P_{i_1}, \ldots, P_{i_k}\}$ con $k \leq K$, vale la seguente proprietà:

- Se $P_{i_j} \in C_i$ mente sul proprio input, la coalizione C_i non è in grado di ricostruire il corretto valore di f in modo autonomo.
- Se ogni P_i fornisce il proprio input corretto ad M, ad ogni P_i viene suggerito il valore corretto di f.

Funzioni k-reversibili

Data $f : T^n \to O$, diremo che f è *k-reversibile*, se, per qualche vettore di k input $t_{C_i} = (t_{i_1}, \ldots, t_{i_k}) \in T$, esistono un altro input $t'_{C_i} = (t'_{i_1}, \ldots, t'_{i_k}) \in T$ ed una funzione g tali che:

- (a) $\forall t_{-C_i} \in T$ abbiamo che $g(f(t'_{C_i}, t_{-C_i}), t_{C_i}) = f(t_{C_i}, t_{-C_i})$;
- (b) per qualche t_{-C_i} abbiamo che $f(t'_{C_i}, t_{-C_i}) \neq f(t_{C_i}, t_{-C_i})$.

Teoremi Ashlagi-Klinger-Tenneholtz

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- *Teorema: Funzione booleana n-NCC:* Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è n-NCC se e solo se è $(n-1)$-NCC.
- *Corollario: Funzioni booleane K-NCC e fortemente-NCC*
  - Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è K-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k \leq K$.
  - Una funzione booleana è fortemente-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k < n$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC se e solo se è NCC.
- Determinare se una data funzione è K-NCC è un problema computazionalmente difficile: determinare se una funzione è dominata è un problema NP-difficile, così come lo è anche il determinare se una funzione è 1-reversibile.

Teoremi Ashlagi-Klinger-Tenneholtz

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- *Teorema: Funzione booleana n-NCC:* Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è n-NCC se e solo se è $(n-1)$-NCC.
- *Corollario: Funzioni booleane K-NCC e fortemente-NCC*
  - Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è K-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k \leq K$.
  - Una funzione booleana è fortemente-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k < n$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC se e solo se è NCC.
- Determinare se una data funzione è K-NCC è un problema computazionalmente difficile: determinare se una funzione è dominata è un problema NP-difficile, così come lo è anche il determinare se una funzione è 1-reversibile.

Teoremi Ashlagi-Klinger-Tenneholtz

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- *Teorema: Funzione booleana n-NCC:* Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è n-NCC se e solo se è $(n-1)$-NCC.
- *Corollario: Funzioni booleane K-NCC e fortemente-NCC*
  - Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è K-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k \leq K$.
  - Una funzione booleana è fortemente-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k < n$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC se e solo se è NCC.
- Determinare se una data funzione è K-NCC è un problema computazionalmente difficile: determinare se una funzione è dominata è un problema NP-difficile, così come lo è anche il determinare se una funzione è 1-reversibile.

Teoremi Ashlagi-Klinger-Tenneholtz

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- *Teorema: Funzione booleana n-NCC:* Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è n-NCC se e solo se è $(n-1)$-NCC.
- *Corollario: Funzioni booleane K-NCC e fortemente-NCC*
  - Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è K-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k \leq K$.
  - Una funzione booleana è fortemente-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k < n$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC se e solo se è NCC.
- Determinare se una data funzione è K-NCC è un problema computazionalmente difficile: determinare se una funzione è dominata è un problema NP-difficile, così come lo è anche il determinare se una funzione è 1-reversibile.

Teoremi Ashlagi-Klinger-Tenneholtz

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- *Teorema: Funzione booleana n-NCC:* Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è n-NCC se e solo se è $(n-1)$-NCC.
- *Corollario: Funzioni booleane K-NCC e fortemente-NCC*
  - Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è K-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k \leq K$.
  - Una funzione booleana è fortemente-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k < n$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC se e solo se è NCC.
- Determinare se una data funzione è K-NCC è un problema computazionalmente difficile: determinare se una funzione è dominata è un problema NP-difficile, così come lo è anche il determinare se una funzione è 1-reversibile.

Teoremi Ashlagi-Klinger-Tenneholtz

- *Teorema: Ashlagi-Klinger-Tenneholtz*: Una funzione booleana è K-NCC se e solo se è $(K-1)$-NCC ed è non-k-reversibile per $k = K$.
- *Teorema: Funzione booleana n-NCC:* Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è n-NCC se e solo se è $(n-1)$-NCC.
- *Corollario: Funzioni booleane K-NCC e fortemente-NCC*
  - Una funzione booleana $f : \{0,1\}^n \to \{0,1\}$ è K-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k \leq K$.
  - Una funzione booleana è fortemente-NCC se e solo se è non-dominata e non-k-reversibile per ogni $1 \leq k < n$.
- Funzioni simmetriche ad n variabili che non sono 1-reversibili non sono nemmeno k-reversibili per qualunque $1 < k < n$; dunque una funzione booleana simmetrica ad n variabili è fortemente-NCC se e solo se è NCC.
- Determinare se una data funzione è K-NCC è un problema computazionalmente difficile: determinare se una funzione è dominata è un problema NP-difficile, così come lo è anche il determinare se una funzione è 1-reversibile.

Equilibrio di Nash K-resistente

Miglior risposta di gruppo

Sia $G = (N, A, u)$ un gioco strategico, dato un sottoinsieme non vuoto di k giocatori che formano una coalizione $C_i = \{P_{i_1}, \ldots, P_{i_k}\} \subset N$, diremo che il vettore di strategie $s^*_{C_i} = (s^*_{i_1}, \ldots, s^*_{i_k})$ è *la miglior risposta di gruppo per* C_i *a* s_{-C_i}, vettore di strategie dei giocatori che non appartengono alla coalizione C_i, se, per ogni vettore di strategie di deviazione correlate $s_{C_i} \in S_{C_i}$ (S_{C_i} è l'insieme dei possibile vettori di strategie per la coalizione C_i) e per tutti i $P_{i_j} \in C_i$, abbiamo che $u_{i_j}(s^*_{C_i}, s_{-C_i}) \geq u_{i_j}(s_{C_i}, s_{-C_i})$.

Equilibrio di Nash K-resistente

Equilibrio di Nash K-resistente

Sia $G = (N, A, u)$ un gioco strategico, diremo che il vettore di strategie indipendenti $s^* = (s_i^*, \ldots, s_n^*) = (s_{C_i}^*, s_{-C_i}^*)$ è un *equilibrio di Nash K-resistente* se, per ogni coalizione $C_i \subset N$ con $|C_i| \leq K$, il vettore di strategie dei giocatori aderenti alla coalizione $s_{C_i}^*$ è la *miglior risposta di gruppo per C_i a $s_{-C_i}^*$*: in sostanza se nessun appartenente a C_i beneficia di qualunque altra scelta strategica $s_{C_i} \neq s_{C_i}^*$.

Equilibrio correlato K-resistente

- Dato un gioco mediato $G = (N, A, u)$, una distribuzione $\mathcal{M} \in \Delta(A)$ è un *equilibrio correlato ex post K-resistente* (con $1 \leq K < n$) se, per ogni coalizione $C_i = \{P_{i_1}, \ldots, P_{i_k}\} \subset N$ con $|C_i| \leq K$, per una qualunque funzione di deviazione di coalizione $r_{C_i} : A_{C_i} \to A_{C_i}$ e per ogni $P_j \in C_i$, vale che

$$u_j(\mathcal{M}) \geq u_j(r_{C_i}(a^*_{C_i}), a^*_{-C_i})$$

(dove $a^* = (a^*_1, \ldots, a^*_n) = (a^*_{C_i}, a^*_{-C_i})$ è un campione di azioni selezionato in accordo con $\mathcal{M}$).

Gioco esteso tramite cheap-talk

Dato un gioco (ad informazione completa oppure incompleta) G ed un protocollo di comunicazione π, è sempre possibile definire un gioco G^* costituito dalle seguenti fasi:

1. *Fase di cheap-talk.* I partecipanti al gioco si scambiano messaggi seguendo il protocollo π in un appropriato *modello di comunicazione*.
2. *Fase di gioco.* I giocatori partecipano al gioco originale G.

Gioco esteso tramite cheap-talk

Dato un gioco (ad informazione completa oppure incompleta) G ed un protocollo di comunicazione π, è sempre possibile definire un gioco G^* costituito dalle seguenti fasi:

1. *Fase di cheap-talk.* I partecipanti al gioco si scambiano messaggi seguendo il protocollo π in un appropriato *modello di comunicazione*.
2. *Fase di gioco.* I giocatori partecipano al gioco originale G.

I payoff dei giocatori in G^* saranno identificati unicamente dai payoff che i giocatori ottengono in G.

Gioco esteso tramite cheap-talk

Dato un gioco (ad informazione completa oppure incompleta) G ed un protocollo di comunicazione π, è sempre possibile definire un gioco G^* costituito dalle seguenti fasi:

1. *Fase di cheap-talk.* I partecipanti al gioco si scambiano messaggi seguendo il protocollo π in un appropriato *modello di comunicazione*.
2. *Fase di gioco.* I giocatori partecipano al gioco originale G.

La strategia s_i di un giocatore P_i nel gioco G^* verrà identificata dalla strategia che questi segue nella fase di cheap-talk seguita dalla scelta di un'azione a_i da giocare in G.

Gioco esteso tramite cheap-talk

Dato un gioco (ad informazione completa oppure incompleta) G ed un protocollo di comunicazione π, è sempre possibile definire un gioco G^* costituito dalle seguenti fasi:

1. *Fase di cheap-talk.* I partecipanti al gioco si scambiano messaggi seguendo il protocollo π in un appropriato *modello di comunicazione*.
2. *Fase di gioco.* I giocatori partecipano al gioco originale G.

G^* sarà detto *gioco esteso tramite cheap-talk*.

Equilibrio di Nash computazionale

È un profilo di strategie indipendenti $s^* = (s_1^*, \ldots, s_n^*)$ in cui ogni strategia s_i^* è *efficiente in* λ tale che, per ogni P_i e per ogni altra strategia s_i *efficiente in* λ, $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) - \epsilon(\lambda)$, dove ϵ è una funzione trascurabile.

Implementazione K-resistente di un mediatore

- G_M versione mediata tramite M di un gioco G.
- s_M profilo a *strategie oneste* ed *equilibrio correlato (ex post) K-resistente* in G_M.
- G_{MPC} versione estesa tramite cheap-talk di G con π protocollo MPC, con partecipanti illimitati/limitati computazionalmente.
- s_π profilo strategico in G_{MPC}.
- Diremo che π è *implementazione K-resistente* di M se:
  - (a) s_π è *equilibrio di Nash regolare/computazionale K-resistente* in G_{MPC};
  - (b) per ogni P_i vale $u_i(s_M) = u_i(s_\pi)$.

Implementazione K-resistente di un mediatore

- G_M versione mediata tramite M di un gioco G.
- s_M profilo a *strategie oneste* ed *equilibrio correlato (ex post) K-resistente* in G_M.
- G_{MPC} versione estesa tramite cheap-talk di G con π protocollo MPC, con partecipanti illimitati/limitati computazionalmente.
- s_π profilo strategico in G_{MPC}.
- Diremo che π è *implementazione K-resistente* di M se:
  - (a) s_π è *equilibrio di Nash regolare/computazionale K-resistente* in G_{MPC};
  - (b) per ogni P_i vale $u_i(s_M) = u_i(s_\pi)$.

Implementazione K-resistente di un mediatore

- G_M versione mediata tramite M di un gioco G.
- s_M profilo a *strategie oneste* ed *equilibrio correlato (ex post) K-resistente* in G_M.
- G_{MPC} versione estesa tramite cheap-talk di G con π protocollo MPC, con partecipanti illimitati/limitati computazionalmente.
- s_π profilo strategico in G_{MPC}.
- Diremo che π è *implementazione K-resistente* di M se:
  - (a) s_π è *equilibrio di Nash regolare/computazionale K-resistente* in G_{MPC};
  - (b) per ogni P_i vale $u_i(s_M) = u_i(s_\pi)$.

Teorema Dodis-Rabin

Siano:

- G_M gioco ad informazione incompleta, versione mediata tramite M di un gioco G,
- s_M profilo a *strategie oneste* ed *equilibrio correlato (ex post) K-resistente* in G_M.
- $(a_1, \ldots, a_n)$ azioni selezionate da M secondo una data distribuzione di probabilità basata sui tipi segreti dei giocatori.
- $f(t_1, \ldots, t_n, r) = (a_1, \ldots, a_n)$ la funzione probabilistica che descrive la strategia di selezione delle azioni raccomandate da M.

Allora:

- è possibile definire il gioco G^*, versione estesa tramite cheap-talk di G, in cui il protocollo π è *implementazione K-resistente* di M se π è un protocollo MPC che computa f in modo sicuro contro coalizioni di al massimo K partecipanti illimitati/limitati computazionalmente.

Corollario Teorema Dodis-Rabin

Sia:

- G_M la versione mediata tramite M di G *gioco di valutazione di* f *funzione* K*-NCC* che ammette al massimo coalizioni di K partecipanti.
- π protocollo MPC che calcola f in modo sicuro contro coalizioni di al massimo K partecipanti illimitati/limitati computazionalmente.

Allora:

- π è implementazione K-resistente di M nel *gioco di Multi-Party Computation* G_{MPC}, versione estesa tramite cheap-talk di G.