Computación segura multiparte coútil para cálculo de funciones arbitrarias

Jesús A. Manjón, Josep Domingo-Ferrer

Universitat Rovira i Virgili, Tarragona, Catalonia
CYBERCAT-Center for Cybersecurity Research of Catalonia

{jesus.manjon, josep.domingo}@urv.cat

Octubre 2022

Outline

1. Introducción
2. Coutilidad
3. Sistema coútil para computación multiparte
4. Análisis de coutilidad
5. Resultados experimentales
6. Conclusiones y trabajo futuro

Introducción

- En la computación segura multiparte (MPC), varias entidades llevan a cabo conjuntamente un cálculo sobre sus respectivos valores de entrada manteniéndolos privados en todo momento
- Desde la aparición de los primeros protocolos en los años 80, y hasta hace pocos años, solo existían protocolos de MPC de uso práctico para ciertas computaciones específicas
- La aparición de compiladores de propósito general para computar funciones arbitrarias es un avance reciente

Introducción

- La mayoría de estos compiladores traducen el código de la computación que se ha de llevar a cabo a un circuito booleano o aritmético y hacen uso de primitivas criptográficas como compartición de secretos, transferencia olvidadiza o circuitos embrollados.
- El uso de los compiladores más avanzados requiere un gran esfuerzo y capacidad por parte del usuario

Proponemos un protocolo MPC con las siguientes características:

- Es de propósito general
- Hace un uso limitado de la criptografía
- Supone la existencia de una comunidad de pares (P2P)
- Se basa en la utilización de un canal anónimo coútil que no depende de una entidad centralizada
- Proporciona resultados correctos y exactos siempre y cuando los participantes sean racionales

Coutilidad

- Un protocolo autoimpuesto (*self-enforcing*) es coútil si su ejecución da lugar a una colaboración beneficiosa para todos los agentes participantes:
  - Un agente que decide participar en el protocolo no se desviará en su ejecución
  - El protocolo es atractivo para todos los agentes
  - No existe otro protocolo que aporte una utilidad mayor al conjunto de agentes ni a un agente concreto

Sistema coútil para computación multiparte

- En nuestro sistema no se pueden vincular de manera inequívoca los valores de entrada y los resultados a sus correspondientes clientes
- No obstante, tanto entradas como resultados pueden ser visibles para otros participantes
- **Si no es aceptable que las entradas o los resultados sean revelados al resto de participantes, nuestro sistema no puede usarse**

Sistema coútil para computación multiparte

Participantes y modelo de seguridad

- Los **clientes** son los participantes que deciden llevar a cabo una computación conjunta. Aportan valores de entrada privados y obtienen resultados privados
- Los **operarios** llevan a cabo los cálculos requeridos para los clientes
- Los **mensajeros** reciben mensajes de los clientes y los reenvían a otros mensajeros o directamente a los operadores
- Los **gestores de responsabilidad** son los encargados de gestionar la reputación de los otros nodos de la red

- Suponemos que los participantes son **racionales**:
  - Siempre y cuando se les proporcionen los incentivos necesarios, los participantes asumirán y cumplirán correctamente con los roles asignados en el protocolo
  - Aun así, se mostrarán interesados en conocer las entradas y los resultados de clientes específicos

- Los protocolos han sido diseñados de manera que los participantes racionales no tengan incentivos reales para desviarse en su ejecución
- Sin embargo, una minoría de participantes maliciosos se comportarán de manera **irracional**: se desviarán de los protocolos aunque haciéndolo obtengan peores resultados globales

- Utilizamos una adaptación coútil del protocolo de gestión de reputaciones descentralizado EigenTrust
- Un cliente **ha de tener una alta reputación** si quiere obtener un resultado correcto de una computación conjunta manteniendo tanto sus valores de entrada como el resultado privados para el resto de participantes
- El incentivo que tiene un nodo para cooperar en el funcionamiento del sistema es el de incrementar su reputación, para así poder convertirse en un cliente exitoso

Para que la reputación sea efectiva se cumplen los siguientes requisitos:

- **Recompensa**. Si un operador lleva a cabo un computación correcta para un cliente, su reputación aumenta
- **Castigo**. Si un operador lleva a cabo de manera incorrecta una computación para un cliente o no sigue los protocolos al pie de la letra, su reputación disminuye
- **Recompensa probabilística**. Un participante que actúa de mensajero tiene una cierta probabilidad de ver aumentada su reputación
- **Utilidad**. Cuanto más alta sea su reputación, más fácil es para un cliente conseguir un resultado correcto a la vez que preservar su privacidad

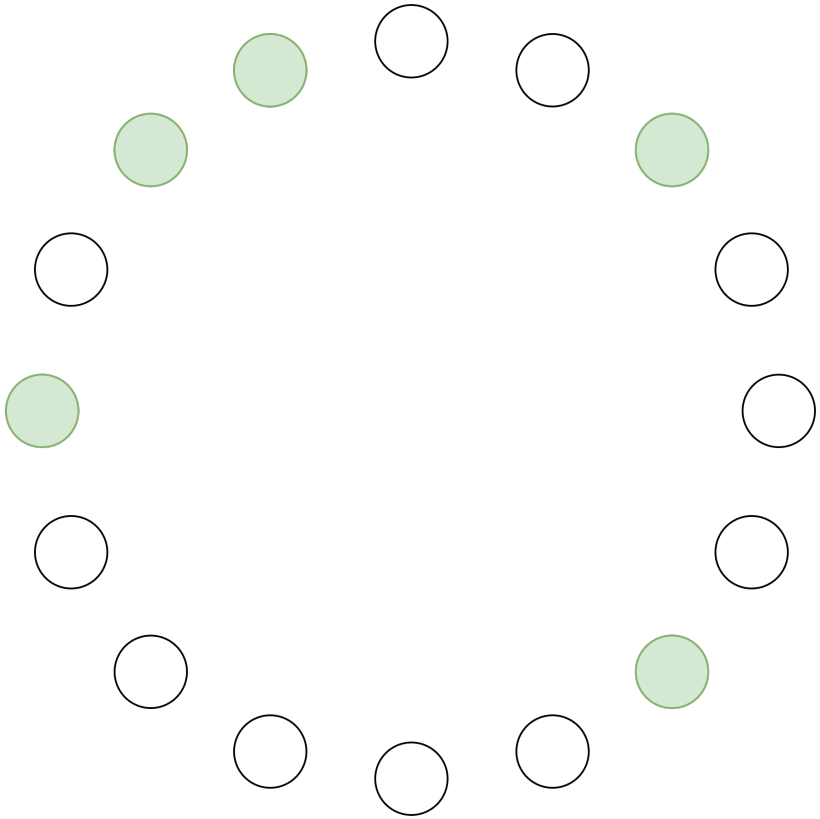
Versión básica, modelo *honest-but-curious*:

- Unos cuantos nodos de la red P2P deciden participar en una computación conjunta y asumen el papel de clientes
- Cada cliente escoge de manera secreta a un operario
- Todos los clientes envían sus entradas a los operarios a través del canal anónimo
- Una vez todas las entradas han llegado a todos los operarios, cada cliente envía a su operario la computación a llevar a cabo y una clave K para que el operario envíe el resultado de vuelta a través de un canal anónimo inverso

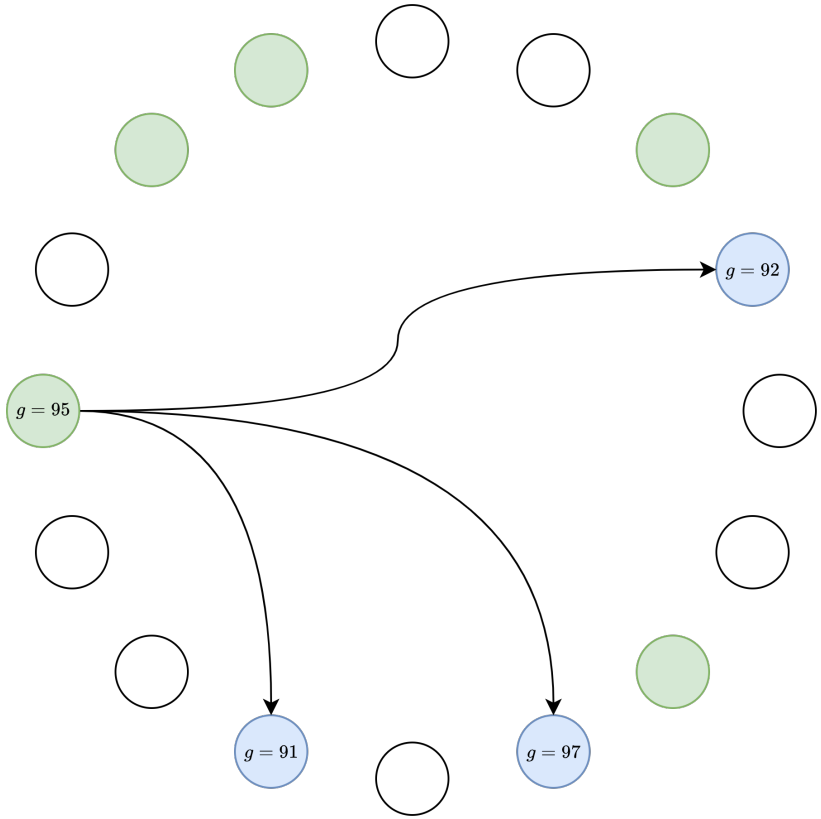
- En el modelo racional los participantes podrían desviarse de su comportamiento esperado
- Se añaden dos mecanismos:
  - **Redundancia**, para detectar una computación incorrecta
  - **Reputación descentralizada**, para recompensar las computaciones y los reenvíos correctos y castigar el mal comportamiento

Sistema coútil para computación multiparte

El protocolo principal



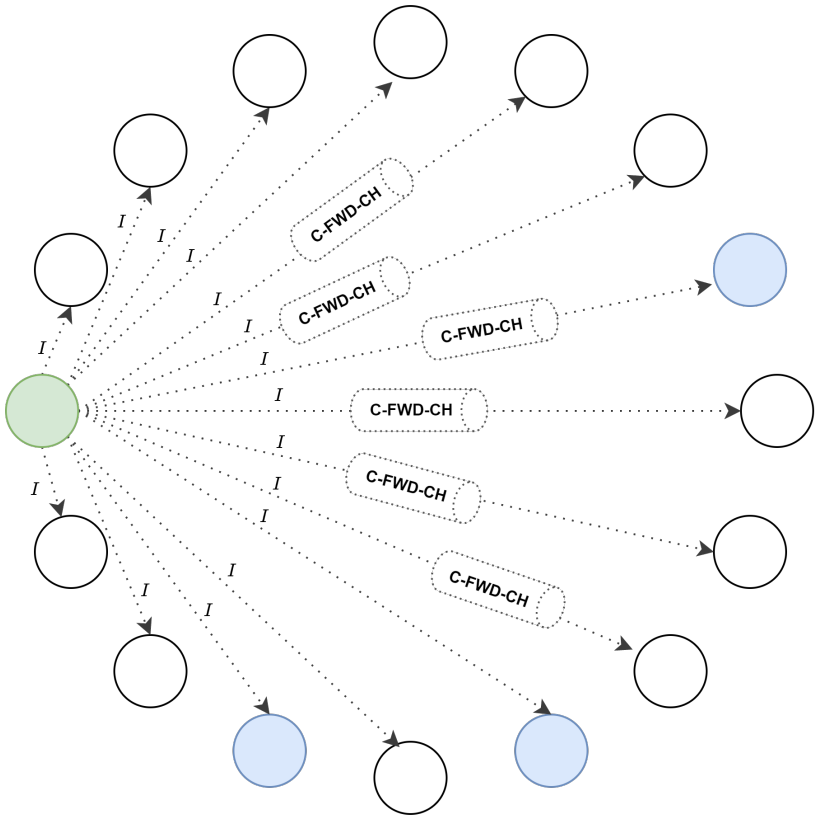
1. n participantes deciden llevar a cabo una computación conjunta



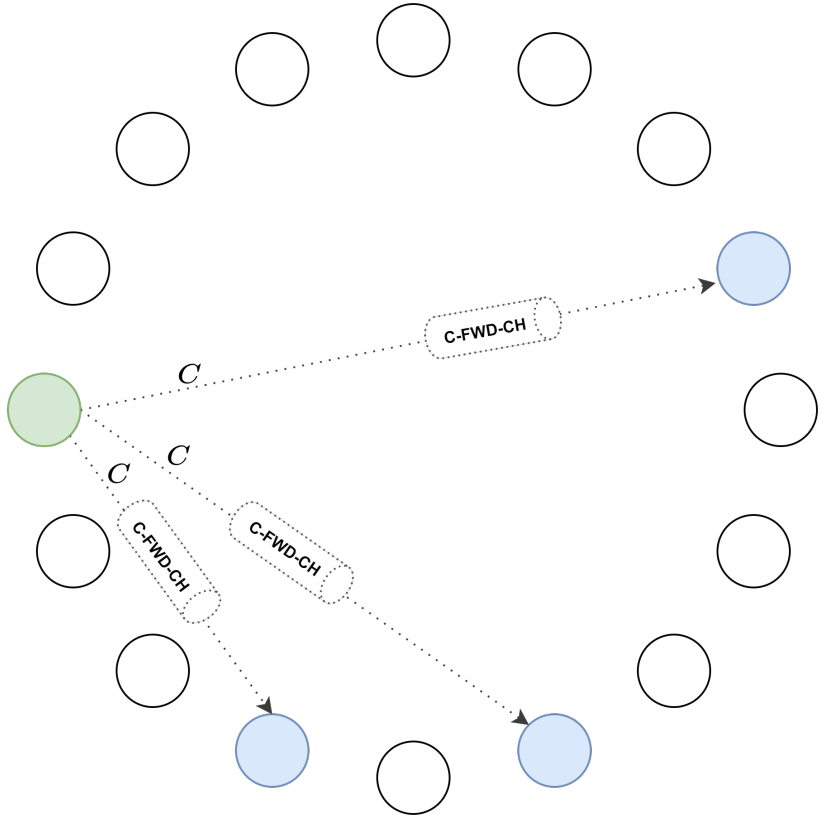
2. Cada cliente escoge varios operarios con **una reputación similar a la suya**

Sistema coútil para computación multiparte

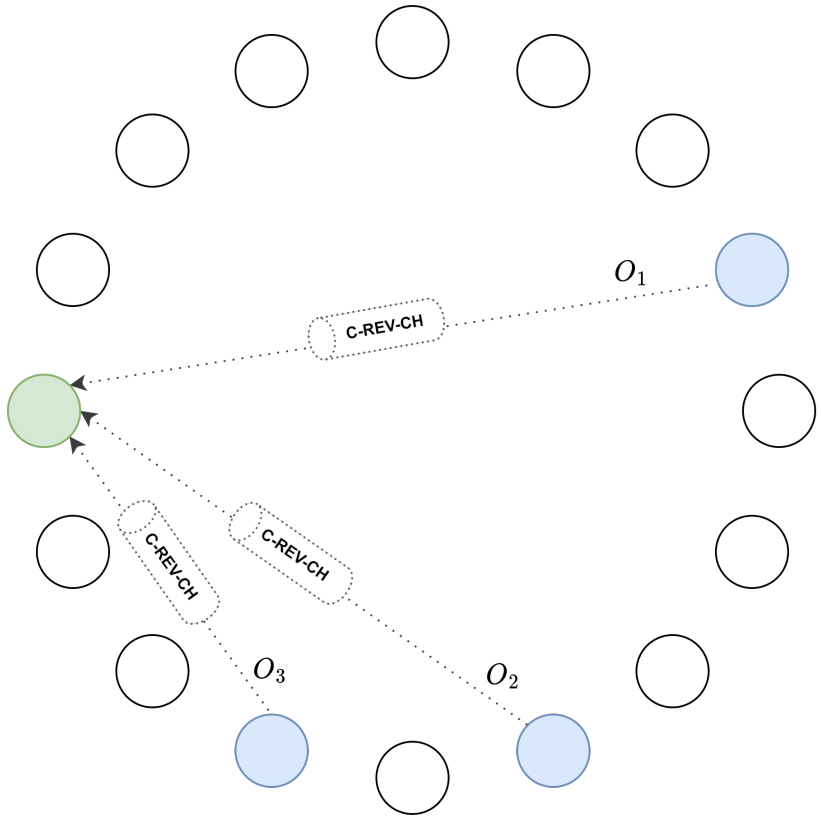
El protocolo principal



3. Todos los clientes envían su entrada a todos los demás nodos de la red a través del canal anónimo



4. Cada cliente envía a sus operarios la operación a llevar a cabo



5. Y espera a recibir el resultado a través del canal anónimo inverso

Sistema coútil para computación multiparte

El protocolo principal

- Una vez el cliente ha recibido todos los resultados, selecciona el mayoritario (más frecuente) y lo considera como válido
- Recompensa con un incremento de reputación a todos los operarios que le han enviado un resultado idéntico al de consenso
- Castiga con un descenso de reputación a todos los operarios que le han enviado un resultado diferente al de consenso
- Cuando todos los clientes han recibido sus resultados, se ejecuta el protocolo que actualiza las reputaciones globales públicas
- Y puede dar inicio otra computación conjunta

Sistema coútil para computación multiparte

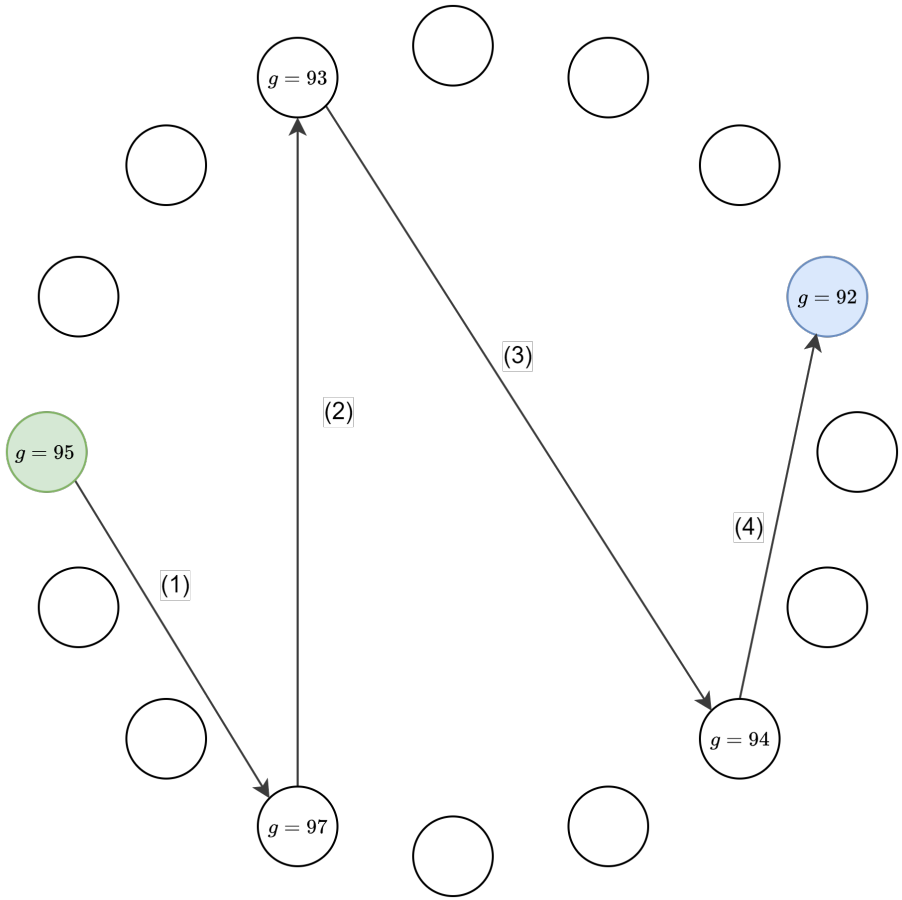
Recién llegados

- Los nodos que entran en el sistema tras su puesta en marcha lo hacen con una reputación $g = 0$, la mínima del sistema
- Durante un número determinado de iteraciones se permite que sean escogidos por los clientes como operarios sea cual sea la diferencia de reputación
- Así el nuevo nodo podrá ganar reputación siempre y cuando ejecute correctamente las computaciones requeridas
- No obstante, no puede ejercer de cliente ni obtener resultados

Sistema coútil para computación multiparte

El canal coútil anónimo

- El nodo emisor escoge aleatoriamente a otro nodo diferente al destino y le envía su mensaje
- Este nodo puede enviar el mensaje a su destinatario o reenviarlo a su vez
- La clave es que un nodo **solo acepta un mensaje si proviene de otro nodo con una reputación muy similar a la suya**



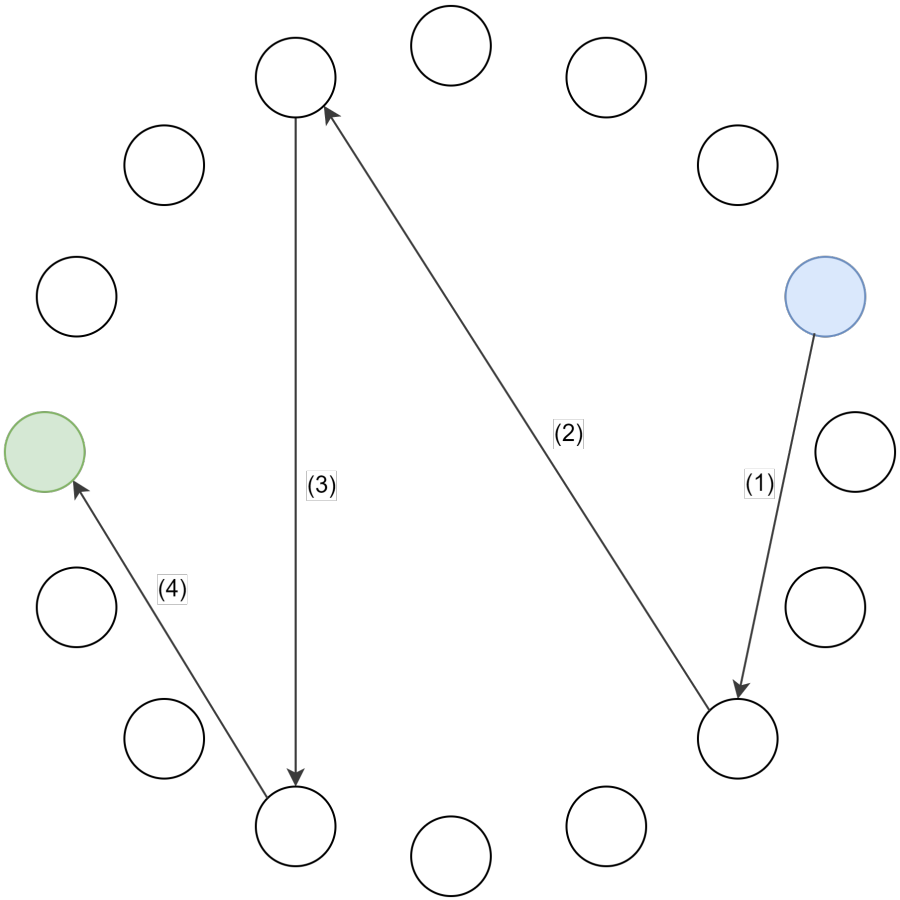
Sistema coútil para computación multiparte

El canal coútil anónimo

- Para definir el concepto de similitud de reputación introducimos el parámetro de flexibilidad δ
- Si un nodo P_i recibe un mensaje de un nodo con una reputación muy inferior (por debajo de $g_i - \delta$) lo descarta
- Por tanto, un nodo con una reputación muy por debajo de la del resto verá como sus mensajes **solo son aceptados por nodos de baja reputación**
- No podrá hacer llegar a un operador fiable (con mucha mayor reputación) la computación a llevar a cabo y no recibirá resultados correctos

Sistema coútil para computación multiparte

El canal inverso



- Una vez se ha llevado a cabo una computación, el operario envía el resultado de vuelta a través del canal inverso
- No puede utilizar el canal coútil anónimo porque **no conoce la identidad del emisor**

Sistema coútil para computación multiparte

El canal inverso

- Se recorre el camino inverso hasta que el resultado llega al cliente, que descifra el resultado al conocer la clave K
- El cliente recompensa al **primer mensajero** con un ligero aumento de reputación

Análisis de coutilidad

Clientes:

- Su objetivo es participar en computaciones conjuntas y obtener resultados correctos, manteniendo privados tanto estos resultados como sus valores de entrada
- Por ello, los clientes llevarán a cabo correctamente sus tareas en los diferenetes protocolos
- Los clientes recompensarán al primer mensajero para evitar el castigo (y ver como se reduce su reputación)
- Tiene sentido que escojan a un recién llegado entre sus operarios: cuanto más nodos honestos participen, más robusto será el sistema y la redundancia minimiza el riesgo en caso de que el recién llegado sea malicioso.

Análisis de coutilidad

Mensajeros:

- Esenciales para la ejecución del canal anónimo y el canal inverso
- Su incentivo es ser recompensados como primeros mensajeros

Operarios:

- Se espera que lleven a cabo la computación requerida
- Su incentivo es ser recompensados si el resultado que devuelven es el mayoritario

Gestores de reponsabilidad:

- Su rol es fundamental para en el cálculo de las reputaciones globales
- Su interés racional es el de favorecer a los nodos que han llevado a cabo de manera correcta sus acciones, ya que ellos mismos pueden ser clientes en futuras rondas de computación

Resultados experimentales

Resultado esperado

- Los nodos honestos han de obtener una mayor proporción de resultados correctos que los nodos maliciosos
- La reputación de cada nodo ha de estar correlacionada con su comportamiento y con la proporción de computaciones correctas que obtiene
- Un recién llegado honesto debería tener a medio plazo una proporción de computaciones correctas parecido a la del resto de nodos honestos
- Un recién llegado malicioso debería mantener una baja proporción de computaciones correctas

Resultados experimentales

Configuración

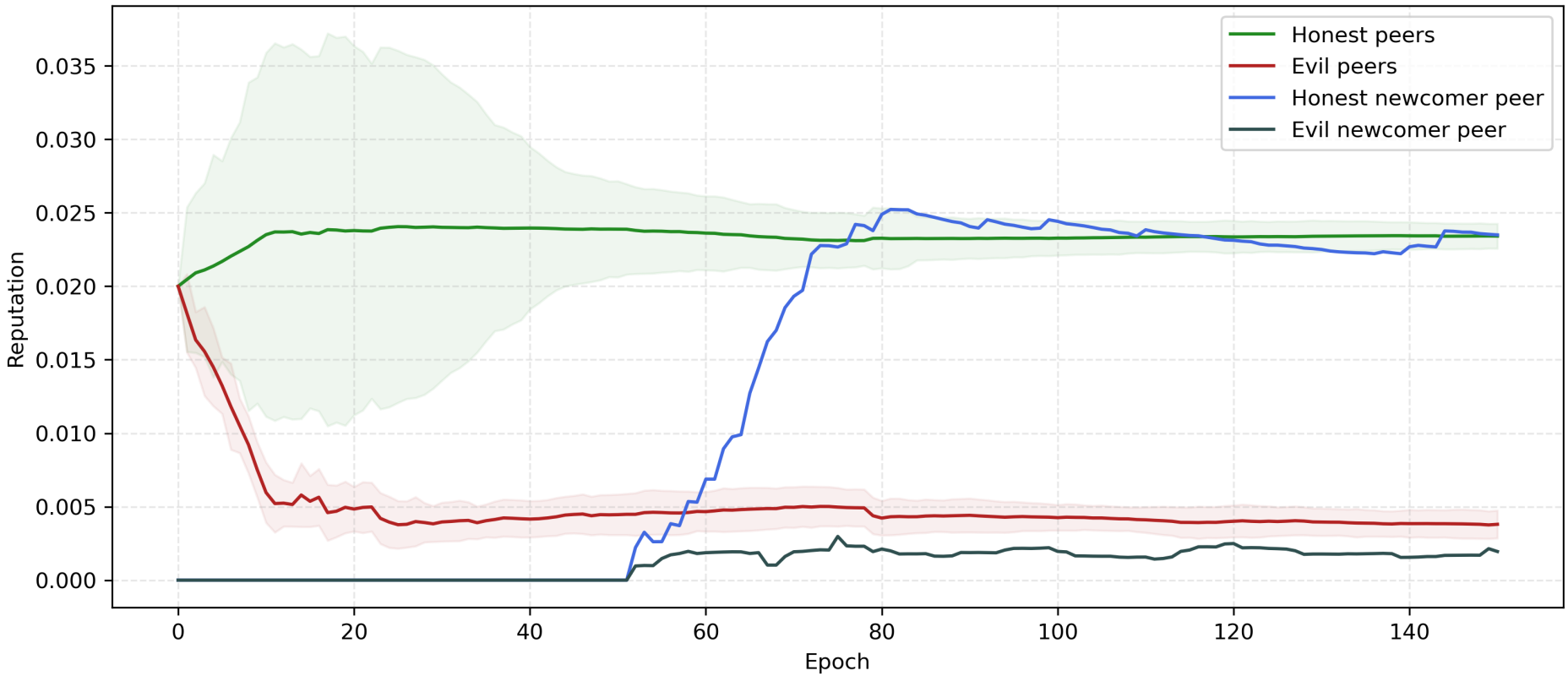
- Red P2P de 50 nodos
- Evolución durante 150 iteraciones
- En cada iteración 10 nodos escogidos aleatoriamente llevan a cabo una computación conjunta
- $r = 3$ operarios redundantes
- Reputación inicial $g = 1/n = 0,02$
- Delta $\delta = g * 0{,}75 = 0{,}015$
- El período para considerar a un nodo como recién llegado es de 25 iteraciones.

Resultados experimentales

Configuración

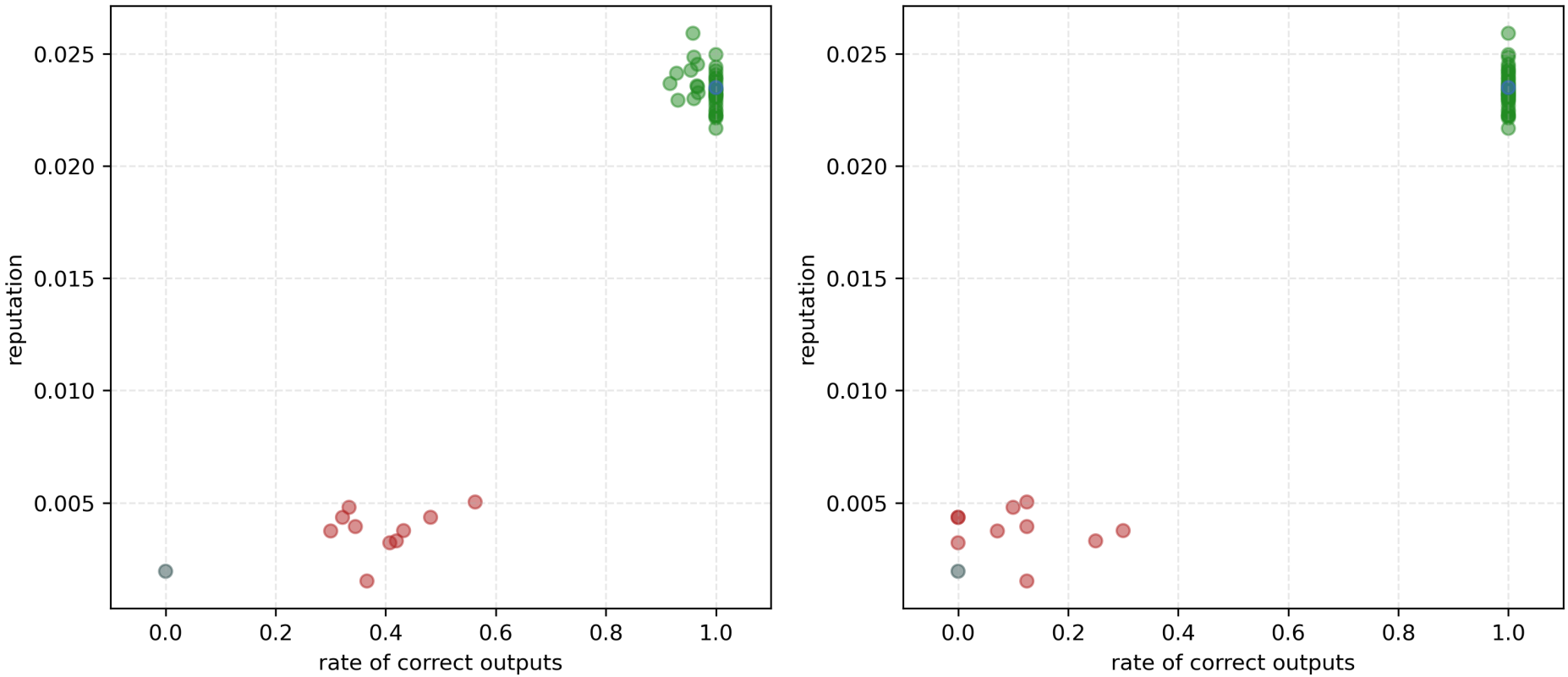
- 20 % de los nodos maliciosos (no computan correctamente para ahorrar sus recursos)
- Se introducen dos nuevos nodos en la iteración 50, uno honesto y otro malicioso

Resultados experimentales



Evolución de la reputación de los nodos de las diferentes clases de nodos

Resultados experimentales



Ratio entre computaciones correctas y reputación de los diferentes nodos. A la izquierda, en las 150 iteraciones; a la derecha, en las últimas 50.

Conclusiones y trabajo futuro

Conclusiones

- Hemos presentado un sistema P2P para MPC con un uso muy moderado de criptografía
- Se asegura que no podrán vincularse inequívocamente los valores de entrada y los resultados con las partes correspondientes
- Es de propósito general

Conclusiones y trabajo futuro

Trabajo futuro

- Ampliar el análisis de coutilidad
- Mayor experimentación para afinar la configuración inicial de los parámetros del sistema
- Reducción del coste de comunicaciones

Conclusiones y trabajo futuro

¡Gracias!