MO829 – Primeiro Semestre 2017
Criptografia e Teoria dos Jogos

Fábio Harada Kubo, RA 116740

Resumo

Criptografia e Teoria dos Jogos são duas áreas da matemática que envolve interações entre múltiplos agentes que, basicamente, não se confiam. Em criptografia, a interação é dada de tal modo que os agentes buscam encontrar maneiras de comunicação para o cálculo de alguma função que toma como entrada dados privados dos agentes, isto é, informações que não podem ser compartilhadas. Enquanto que em teoria dos jogos essa interação é feita na forma de um jogo em que cada jogador adota uma estratégia procurando maximizar sua recompensa, determinada de acordo com as estratégias adotadas por todos os jogadores. Será mostrado aqui noções de criptografia aplicáveis em teoria dos jogos, de modo a tentar encontrar um jogo equivalente a um que possua um mediador confiável, sem a presença desse intermediário.

1 Introdução

Nos últimos anos vimos a emergência de diversas tecnologias que apresentam um ponto em comum: a descentralização do modelo. Nesse modelo, os sistemas não mais apresentam um órgão central do qual se é dependente para tomada de decisões, os próprios participantes se coordenam de modo a assegurar que o sistema funcione plenamente. Mas, como garantir que nenhum nó, ou um grupo de nós, apresente um comportamento fora do padrão e consiga comprometer o sistema? Uma das possíveis formas é o uso de protocolos MPC (*multi-party computation*).

2 Noções de Criptografia

Multi-party computation (MPC) é uma subárea da criptografia que visa a criação de métodos para que n agentes, que não se confiam, consigam calcular corretamente $s = f(t_1, t_2, ..., t_n)$ que depende de informações privadas dos agentes, sem que haja vazamento de informação, isto é, que cada agente não consiga deduzir nenhuma informação além do resultado da função e de sua informação privada. Um exemplo clássico de problema que tenta se resolver é o problema dos milionários de Yao [2]. Nesse problema, dois milionários querem saber quem possui a maior fortuna, sem saber o quanto o outro possui. Esse problema é facilmente resolvível se ambos confiarem em uma terceira pessoa, mas o que um protocolo MPC faz é resolver tal problema sem a necessidade dessa terceira pessoa.

Importante notar que a função pode ser generalizada para outros cenários. Ela pode ser uma função probabilística, nesse caso o resultado da função depende também de uma variável probabilística ($s = f(t_1, t_2, ..., t_n, X)$, onde X é uma variável probabilística). Ou apresentar múltiplas saídas, nesse caso cada saída da função é associada a um participante ($(s_1, s_2, ..., s_n) = f(t_1, t_2, ..., t_n)$). Um exemplo para ambas as generalizações, suponha um leilão onde P_1 é o leiloeiro de um item e P_i ($i \in [n] \backslash \{1\}$) querem comprar tal item atribuindo-lhe um valor t_i.

A função $f(-, t_2, ..., t_n, X)$ calcula o vencedor e o maior valor, informando apenas o leiloeiro $((s_1, s_2, ..., s_n) = ((val, j), 0, ..., 0)$, onde $val = \max(t_2, ..., t_n)$ e $j = arg\max(t_2, ..., t_n))$, realizando o desempate de acordo com um critério de desempate associado a X (note que o valor de X não pode ser controlado por nenhum participante, sendo gerado apenas quando o cálculo é realizado).

Uma das partes mais interessantes de MPC é a elaboração de protocolos que consigam calcular a função, mesmo quando um dos participantes, ou um grupo de participantes, não siga o protocolo corretamente. Participantes que tendem a não seguir o protocolo são chamados de *faulty*, sendo incluido nessa categoria qualquer agente com um comportamento que desvie do que o protocolo exige, ou seja, são incluídos nessa categoria desde os honestos mas curiosos (honestos pois seguem os passos do protocolo corretamente, mas curiosos pois querem saber informações privadas dos outros participantes) até o malicioso, que não mede esforços para alterar o resultado da função, ou até mesmo aprender as informações privadas dos outros participantes. Assumindo que todos os agentes *faulty* se organizam e coordenam um ataque ao protocolo, temos um adversário A. Procura-se, então, protocolos MPC que sejam seguros contra adversários de tamanho $k < n$.

2.1 Definições de Segurança

Para que um protocolo MPC seja considerado seguro, deve-se garantir que as saídas dos agentes honestos sejam corretamente calculadas e que nenhum participante tenha acesso a informação privada de outro participante. Essas duas propriedades são garantidas se existir um agente confiável que recebe as informações dos participantes e calcula a saída de todos, sem informar as entradas para ninguém. Considerando tal situação um modelo ideal, define-se formalmente que um protocolo MPC é seguro se para todo adversário A o resultado que os agentes honestos recebem é indistinguível do resultado do modelo ideal na presença de um adversário A' que controla os mesmos agentes do adversário A. Note que essa definição de segurança permite que a saída de um agente honesto, quando existe um adversário A, ser diferente da saída quando todos são honestos. Ou seja, a noção de corretude é mais relaxada, garantindo-se apenas que todos os agentes honestos recebem seu resultado e ele é idêntico ao que um agente confiável calcularia se recebesse as mesmas entradas.

Garantir que a informação privada de qualquer agente não vaze é primordial em qualquer protocolo MPC (e em qualquer área da criptografia!), assim podemos ter algumas definições mais fracas de segurança em relação as saídas. Uma relaxação da garantia de que todos os agentes honestos receberão sua saída é chamada de *fairness* (justiça). Nesse caso, se algum agente recebe sua saída, então todos os agentes honestos também recebem sua saída. Assim, protocolos justos são aqueles em que ou todos os agentes honestos recebem sua saída ou ninguém recebe.

Uma definição de segurança ainda mais relaxada é a de corretude: se um agente honesto recebe sua saída, então ela é correta (idêntico ao resultado calculado no mundo ideal). Note que em protocolos que garantem apenas corretude (e privacidade), os agentes do adversário podem receber seu resultado e, ainda, prevenir que os agentes honestos recebam o deles.

2.2 Variações na modelagem do sistema

Os conceitos de segurança apresentados são universais, entretanto protocolos MPC dependem, também, do modelo do sistema ao qual ele será implementado. Assume-se que o adversário A consegue controlar no máximo $k < n$ agentes *faulty*, podendo possuir poder computacional ilimitado (nesse caso, o protocolo fornece resultados corretos se eles são estatísticamente indistinguíveis do resultado do mundo ideal) ou limitado. Neste caso, o protocolo MPC é parametrizado por um parâmetro de segurança λ e todos os cálculos e comunicações são realizados em tempo polinomial

em λ. Nesse cenário, o protocolo fornece resultados corretos se os resultados do protocolo e do mundo ideal forem computacionalmente indistinguíveis.

Existem variações, ainda, no modelo de comunicação entre as partes. Quando existe um canal seguro e autenticado entre todas as partes (isto é, todo par de participante consegue trocar informações sem que tal informação seja vazada), disse-se que o modelo possui canais seguros. Um canal de *broadcast* seguro ocorre quando existe um canal no qual toda informação propagada nele chega em todos os participantes sem que sofra alterações. Existem, ainda, mais dois modelos de comunicação: envelopes e urnas. A utilização de envelopes garante que toda comunicação é privada e não pode ser violada. Enquanto que as urnas são uma generalização de envelopes, imagine uma urna onde são colocados envelopes e qualquer ação sobre essa urna é conhecida por todos os participantes. Assim, todos sabem quando alguém escreve um envelope (embora não saibam seu conteúdo), quando alguém escolhe um envelope e revela seu conteúdo e quando alguém destrói um envelope.

2.3 Resultados para protocolos MPC

A depender do modelo de sistema escolhido, garante-se que existe algum protocolo MPC seguro contra adversários de até k agentes. Assumindo que os adversários são computacionalmente limitados, existem canais seguros e um canal de broadcast seguro, então existe um protocolo MPC que garante que todos os agentes honestos receberão sua saída corretamente apenas para $k < n/2$. Se assumirmos apenas corretude e privacidade, então existe um protocolo MPC seguro para qualquer $k < n$. Admitindo o uso de envelopes, então existe um protocolo justo para $k < n$.

Se admitirmos que o adversário é computacionalmente ilimitado, então garante-se que os agentes honestos receberão sua saída corretamente apenas se existirem canais seguros de comunicação e se $k < n/3$. Se assumirmos que existe ainda um canal de broadcast seguro, existe um protocolo seguro para $k < n/2$. Admitindo agora o uso de envelopes, urnas e de um canal de broadcast seguro, então existe um protocolo seguro para qualquer $k < n$.

3 Aplicação de protocolos MPC em Teoria dos Jogos

Consideraremos aqui apenas jogos que existem um mediador confiável que recomenda privadamente a cada jogador uma estratégia de acordo com uma distribuição de probabilidade, cabendo a cada jogador decidir se segue a estratégia ou não. Tal recomendação é feita de tal modo que todo jogador siga a estratégia recomendada, pois é a que lhe garante a maior recompensa esperada, atingindo-se, assim, um equilíbrio correlacionado. Assumiremos, ainda, que antes do começo do jogo existe uma fase preliminar onde todos os participantes podem se comunicar de acordo com algum modelo de comunicação sem afetar as recompensas do jogo original. Tal extensão é chamada de jogo extendido por conversa fiada (*cheap-talk extended game*).

3.1 Novas noções

No cenário proposto, assumiremos que todos os jogadores são computacionalmente limitados. Podemos definir, então, o que é equilíbrio computacional. Se a fase preliminar for modelada de acordo com um problema computacional difícil, então existe a pequena possibilidade de haver um atacante que resolva o problema, obtendo, assim, uma recompensa maior do que a esperada se ele adotar a estratégia do equilíbrio. Surge, então, o conceito de equilíbrio computacional de Nash, onde garante-se que resolver o problema computacional fornece vantagens desprezíveis.

Outro conceito novo é o de k-resiliência. Um equilíbrio é dito k-resiliente se nenhuma coalizão de até k membros consegue se beneficiar se desviarem do equilíbrio. Existem aqui duas

variantes: *ex ante*, onde os membros da coalizão podem se organizar apenas antes de receberem as estratégias do mediador, e a variante *interim*, onde a coalizão pode se organizar após o conhecimento das estratégias fornecidas pelo mediador.

3.2 Removendo o mediador

Dado que a presença de um mediador possibilita que as recompensas dos jogadores sejam maiores (equilíbrio correlacionado é um superconjunto de equilíbrio de Nash), seria interessante pensar em modos de possível remover tal mediador do jogo, já que a existência de um agente confiável e incorruptível não é sempre realizável. A utilização de protocolos MPC é uma possível resposta. Se f é uma função que especifica um equilíbrio correlacionado k-resiliente, então executar um protocolo MPC seguro na fase preliminar, simulando a função do mediador, leva a um jogo com um equilíbrio com as mesmas recompensas dadas para o jogo com o mediador. Pois um protocolo MPC seguro garante que todos os agentes honestos receberão sua saída para um adversário que controla k agentes. Assim, como a k-resiliência garante que nenhuma coalizão de tamanho k possui motivação para desviar do equilíbrio, todas as partes agiram honestamente e as recompensas são as mesmas.

Se permitimos um protocolo MPC justo, devemos assumir que existe uma forma de detectar desvios. Dessa forma, pune-se os agentes que desviaram do protocolo excluindo-os. Eventualmente, o protocolo termina e todos os agentes honestos recebem sua saída, temos assim uma situação idêntica ao do protocolo seguro. Porém, se permitimos apenas um protocolo correto e privado, então só existe solução para $k = 1$, onde todos os jogadores podem optar por punir o agente malicioso.

4 Conclusão

Vimos aqui que, embora criptografia e teoria dos jogos possuam objetivos diferentes, podemos utilizar conceitos de uma área para desenvolvimentos em outra. Apresentou-se aqui apenas uma aplicação (remoção de mediador em jogos extendidos por conversa fiada), entretanto podemos fazer o caminho inverso e procurar por protocolos MPC que assumem que os agentes são racionais, agindo, portanto, por interesses próprios. Ou seja, da interação entre esses dois campos podemos entender melhor a natureza de interação entre agentes na presença de conflito de interesses.

Referências

[1] Y. Dodis and T. Rabin. Cryptography and game theory. In N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 8, pages 181–204. Cambridge University Press, New York, NY, USA, 2007.

[2] A. C. Yao. Protocols for secure computations. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 00:160–164, 1982.