Pruebas de conocimiento cero

Andrea González Luis Mayo Carlos Acosta

12 de junio de 2017

Índice

1. Introducción

En nuestras experiencias tanto dentro como fuera del mundo académico, por lo general se nos exige que cualquier proposición comunicada al otro, sea sustentada con evidencias claramente expuestas como acompañamiento de nuestra declaración. A fin de inducir en ella un carácter de verdad, la defensa de nuestros enunciados debe apoyarse de argumentos que conserven la validez dentro del objetivo que perseguimos.

Pero, ¿qué ocurre cuando no podemos permitir que nuestra audiencia se entere de los detalles del camino que nos condujo al resultado que le presentamos?, ¿es posible mantener la confiabilidad, sin necesidad de proveer información más allá de la afirmación de que lo que decimos es verdad?

Sin cruzar apresuradamente al terreno de la magia o de la fe, consideremos primero la noción de las ***pruebas de conocimiento cero*** (ZPK[1]). Imaginemos que estamos solicitando un trabajo para una organización o empresa y es necesario que les convenzamos de nuestra valía, sin embargo toda la experiencia con la que contamos ha sido obtenida en los círculos del bajo mundo o la clandestinidad y no estamos en libertad de proporcionar un *curriculim vitae* o documento que demuestre nuestra competencia. En ese caso la empresa podría someternos a un periodo de prueba que consista en resolver problemas o tareas que atañen a nuestras habilidades, aún sin conocer el proceso que estamos llevando a cabo para ninguna de ellas -ya que no podemos revelar dichas técnicas por su naturaleza secreta-, entre más labores sean requeridas más certeza tendrán de aceptar nuestra solicitud de trabajo y menor probabilidad de que no estemos siendo honestos.

Así, en una prueba de conocimiento cero, si A (el "demostrador") busca probar a B (el "verificador") que una proposición X es verdadera, al término del proceso A estará completamente convencida de X, pero no habrá obtendido ningún nuevo conocimiento (barak []). De ahí el nombre, propuesto por primera vez en 1985 por los científicos de la computación Shafi Goldwasser, Silvio Micali, y Charles Rackoff en su artículo *La complejidad del conocimiento de los sistemas de demostración interactivos* [2], en el que definieron una nueva jerarquía de para las pruebas de conocimiento interactivas -de las que hablaremos más adelante-; concibieron además el concepto de *complejidad de conocimiento*[3] y presentaron el primer ejemplo de una prueba de conocimiento cero para un problema concreto. Sus esfuerzos en esta materia les valió a los tres autores un *premio Gödel* en 1993[4].

[1] Del inglés: *Zero Knowledge Proofs*

[2] Intento de traducción del inglés: *The knowledge complexity of interactive proof systems*

[3] Del inglés: *Knowledge complexity*, medida de la cantidad de conocimiento que el demostrador transfiere al verificador

[4] Premio anual otorgado por EATCS y la ACM SIGACT para artículos destacados en teoría de las ciencias de la computación. Lista de Ganadores del premio Gödel

1.1. Una pequeña historia

En aras de entrar definitivamente en el tema de las pruebas de conocimiento cero, nos auxiliaremos de un ejemplo que aunque ficticio, bastante didáctico, inspirado en la clásica historia de *Alí Babá y los cuarenta ladrones*[5].
En esta adecuación del cuento, presentada en el artículo *Cómo explicar a tus hijos los protocolos de conocimiento cero*[3], Alí Babá se encuentra, como todos los días, en el bazar de su pueblo al este de Bagdad, cuando de pronto es atracado y separado de sus pertenencias por un ladrón. Alí Babá sigue al malhechor hasta una cueva cuya entrada es tan obscura que lo pierde de vista. Al explorar su interior, se enfrenta a una encrucijada y debe tomar uno de dos caminos. Al elegir el de la derecha, se encuentra con una pared al final de éste, sin hallar al ladrón; regresando sobre sus pasos, camina sobre el camino que rechazó inicialmente hasta llegar a su pared final respectiva, sin pista del ladrón todavía. Decide entonces que tomó la decisión incorrecta y que gracias a ello el ladrón tuvo el tiempo de salir de la cueva y escapar. Sin embargo, lo mismo ocurre durante los siguientes 39 días, y aún procurando alternar su decisión para hallar al responsable de su desgracia, "fallando" una vez tras otra. Es entonces que decide revelar el misterio dentro de la cueva y entra desde temprano por uno de los caminos, esperando por la llegada de uno de los ladrones a la pared de su elección. Por fin presencia a uno de ellos escapar de ese punto a lo que parecía el otro camino por un pasaje secreto en la pared, con sólo murmurar las palabras mágicas: ábrete sésamo. Luego de su hallazgo, se dedica a cambiar la clave por una propia para terminar con los robos en el bazar de su pueblo. Compartiría esta palabra a sus descendientes únicamente, para que su secreto y su buena acción no muriesen con él.
Supongamos ahora que Pablo, a pesar de tantas generaciones intermedias, clama conocer las palabras mágicas de la cueva. Carlos, un entusiasta del estudio de la genealogía de Alí Babá, quien no reconoce a Pablo entre los descendientes de Alí Babá, le pide que demuestre su conocimiento sobre el secreto de su antecesor. Pablo naturalmente se niega, pues debe proteger años de esfuerzo por mantener las palabras mágicas dentro de su línea sanguínea. Afortunadamente, antes de darse por vencidos, consultan con una estudiante del curso de Criptografía en una mítica Facultad de Ciencias por un método o procedimiento que les permita probar que Pablo dice la verdad. Andrea les comenta sobre las pruebas de conocimiento cero y cómo pueden servir para lograr su objetivo. Luego de hacer los preparativos para su expedición, marchan juntos hasta la cueva misteriosa del relato. Estando ahí, Andrea pide a Pablo (el demostrador) que entre a la cueva y sin que el resto lo pueda presenciar, lance una moneda y si le toca águila vaya por el camino derecho hasta el fondo, o por el camino izquierdo en caso contrario. Luego de dar unos momentos a Pablo para llegar a su destino, a la al-

[5]Perteneciente a la recopilación de *Las mil y una noches*

tura de la encrucijada, Andrea pide a Carlos (el verificador) que lance su propia moneda y le grite a Pablo el resultado. Pablo tendrá que regresar por el camino de la derecha si toca águila o por el camino de la izquierda si toca sol. Después de repetir esto 40 veces -en honor a su antecesor- con éxito, Pablo demuestra a Carlos que conoce el secreto sin necesidad de revelarlo.
Con un pequeño análisis de probabilidad, el fundamento de la prueba puede ser explicado fácilmente. Si Pablo dice la verdad, puede regresar de cualquiera de los caminos por el lado que le piden, con probabilidad 1, sea necesario usar la palabra mágica o no, las veces que sean. En caso de que mienta, sólo podrá regresar por el camino que le piden con probabilidad[6]

$$Pr[sol_{m2}|sol_{m1}] = Pr[aguila_{m2}|aguila_{m1}] = 1 \cdot \frac{1}{2} = \frac{1}{2} \tag{1}$$

pues Pablo podría regresar del lado correcto sólo si desde el inicio tomó ese camino. Luego de las 40 repeticiones del proceso en la cueva, la probabilidad de fallo de Pablo estará dada por

$$1 - \frac{1}{2^{40}} \approx 0.9999999999990905 \tag{2}$$

Pues también las repeticiones son independientes entre sí. En este punto, Carlos estaría más que convencido (probabilísticamente, al menos) de que Pablo dice la verdad si lo logra las 40 veces, -aunque con 8 repeticiones bastaría para alcanzar la probabilidad de 99 % de fallo si Palo no es honesto.

En este ejemplo también se muestra una de las propiedades de las pruebas de conocimiento cero, que refiere a la imposibilidad de que Carlos convenza a un tercero de que Pablo conoce el secreto, dejando más claro la utilidad de que el verificador no obtenga conocimiento. Pero no nos adelantemos, primero un par de características más antes de tocar ese punto.

1.2. Características

2. Sistemas de Demostración Interactivos

La clase de complejidad *IP* [1]

3. Instancia

3.1. Protocolo de conocimiento cero para *logaritmo discreto*

[4] Sea $G = \langle g \rangle$ un grupo cíclico de orden q con un generador g, ambos q y g conocidos, y sea $x \in G$ un elemento arbitrario del grupo que tiene el logaritmo

[6] Dado que el lanzamiento de las monedas son eventos independientes.

discreto $w = log_g(x)$. Tanto el demostrador P como el verificador V reciben como entrada a x, pero P recibe además a w.
El protocolo entre P y V se describe a continuación.

1. $P(x, w)$ elige aleatoriamente a un elemento $0 \leq r < q - 1$ y envia $z = g^r (\text{mod } q)$ a V.
2. $V(x)$ envia un bit $b \in \{0, 1\}$ aleatorio a P.
3. P responde con $a = r + b \cdot w \ (\text{mod } q)$
4. V acepta si $g^a = z \cdot x^b \ (\text{mod } q)$

La idea básica es que si el bit $b = 1$, entonces P envia un número que parece aleatorio $(a = r + b \cdot x (\text{mod } q))$ a V, pero V ya conoce $z = g^r (\text{mod } q)$ y sabe que $x = g^w$ por lo que puede multiplicar estos y compararlos con g^a.
En realidad, V solo puede ver a z y a y lo que sabe es que $a = log_g(z) + w$. Como ambos conocen s, pero el demostrador además conoce a w, entonces solo le queda demostrarle al verificador que también sabe $log_q(z)$.
Ahí es donde entra el bit aleatorio que envió V. Si $b = 0$, P solo envia $s = r$ de vuelta a V en el paso 3. V revisa que $z = g^r (\text{mod } q)$, es decir, $r = log_q(z)$. De este modo, dependiendo del valor de b, el verificador obtendrá r o a pero jamás ambas (ya que su diferencia es precisamente w). Por lo tanto V no obtiene ninguna información acerca de w.

3.1.1. Análisis

Veamos entonces que el protocolo satisfaga las tres propiedades de las pruebas de conocimiento cero.

- **Totalidad:**
  Si P y V actuan como está descrito en el protocolo, entonces tenemos que

$$g^a = g^{r+b \cdot w} = g^r \cdot (g^w)^b = z \cdot x^b$$

- **Solvencia:**
  Hay que notar que todas las $x \in G$ tiene logaritmo discreto, entonces no puede haber ningún demostrador deshonesto que engañe al verificador de que su declaración es cierta cuando sea falsa. Podríamos decir que el concepto de *solvencia* no es particularmente significativo en este caso.

- **Conocimiento cero:**
  Supongamos que existe un verificador engañoso V^*. Definimos entonces un simulador $mathcalS^{V^*}(x)$ del modo siguiente:

  1. Elige un bit aleatorio b y un elemento $a \in G$.

2. Envia $z = g^a/x^b$ a V^* y recupera un bit de desafío b^*. Si V^* responde con un mensaje mal formado o aborta, solo muestra la salida de la vista hasta el momento.
3. Si $b^* = b$, completa la vista usando a a como el último mensaje del demostrador, en caso contrario rebobina V^* y repite la simulación.

   Es relativamente fácil demostrar que S reproduce la vista de V^* hasta una distancia estadística insignificante ya que el valor de z calculado por S es estadísticamente independiente de su bit b, por lo tanto, tenemos que

$$Pr[b^* = b] = \frac{1}{2}$$

3.2. Otras aplicaciones

3.2.1. zk-SNARK

Las zk-SNARKs (zero knowledge Succint Non-Interactive Arguments of Knowledge) son herramientas criptográficas que consisten en pequeñas pero bien definidas pruebas de conocimiento cero que además de ser fáciles de verificar, no revelam ninguna información y no existe iteracción alguna entre demostrador y verificador. Podemos considerarlas como pequeños circuitos lógicos que necesitan generar una prueba de una sentencia para verificar cada una de las transacciones. La manera en la que logran esto es al tomar una muestra de cada transacción, generan una prueba y luego intentan convencer al receptor de que los calculos fueron hechos correctamente sin revelar ninguna información que no sea la prueba misma. La operación básica de una ejecución SNARK es una entrada códificada en este circuito que puede descifrarse.
La ventaja de que las zk-SNARKs puedan ser verificadas rápidamente y las pruebas son cortas, es que pueden proteger la integridad del cálculo sin sobrecargar a los no participantes. Tienen como desventaja su falta de escalabilidad pues son muy intensivas en uso de CPU para generar pruebas y les toma hasta 1 minuto generar una nueva prueba.
Algunos de los protocolos que usan estas herramientas son *Zcash* y *Hawk*.

3.2.2. Zcash

Zcash es una criptomoneda descentralizada y de código abierto que ofrece privacidad y transparencia selectiva de transacciones. Si bien los pagos de Zcash son publicados en una cadena de bloques pública, tanto el remitente como el destinatario y el monto de la transacción permanecen privados.
A grandes rasgos. podemos considerar a *Zcash* como una extensión del protocolo de *Bitcoin* pues agrega algunos campos al formato de transacciones de *Bitcoin*

con la finalidad de que soporte transacciones cifradas. *Zcash* usa zk-SNARKs para cifrar todos sus datos y solo otorga llaves de descifrado a las partes autorizadas a ver tales datos.
Hasta el lanzamiento de *Zcash*, en cadenas de bloques públicas no era posible cifrar todos los datos, pues eso impedía que los denominados "mineros"de la cadena pudieran verificar si las transacciones eran válidas o no. Fue gracias a las pruebas de conocimiento cero que se le permitió al creador de una transacción poder probar que la transacción era válida sin que se le revelara la dirección del emisor, la dirección del receptor y el monto de la transacción.
Zcash también permite a los usuarios enviar pagos públicos que funcionan de forma similar a *Bitcoin*. Con el soporte de direcciones tanto blindadas como transparentes, los usuarios pueden elegir enviar *Zcash* en privado o en público. Los pagos de *Zcash* enviados desde una dirección blindada a una dirección transparente revelan el saldo recibido, mientras que los pagos desde una dirección transparente a una dirección blindada blindan el valor recibido.

3.2.3. Hawk

Hawk es un framework para generar contratos inteligentes que conserven la privacidad de los usuarios, funcionando de una manera similar a *Zcash*.
Hawk no almacena las transacciones financieras en la cadena de bloques,sino que se encarga de mantener confidenciales tanto el código del contrato, como la información que le fue enviada y el monto de dinero que se envió y se recibió, permitiendo que lo único que sea visto sea la prueba.
Mientras que la privacidad en cadena protege a ambas partes contractuales contra el público (aquellos no involucrados en el contrato finaciero) , la seguridad contractual protege a las partes involucradas entre sí.
Hawk asume que ambas partes contractuales actuan egoístamente para maximizar y beneficiar sus propios intereses financieros. En particular, pueden desviarse arbitrariamente del protocolo prescrito o incluso abortar prematuramente. Por lo tanto, la seguridad contractual es una noción multifacética que abarca no solo las nociones criptográficas de confidencialidad y autenticidad, sino también la equidad financiera en presencia de comportamientos de engaño y abandono.
Para cada contrato, de manera similar con *Zcash*, hay también paramétros públicos confiables. La única manera de generar estos paramétros es mediante un proceso que involucra generar un valor secreto en cada paso intermedio, el cual debe borrarse al final del protocolo. Aún no hay una fecha de lanzamiento anunciada por parte de sus desarrolladores pues continuan trabajando en optimizaciones de sus herramientas de compilación para SNARK con el fin de mejorar el rendimiento de sus protocolos.

4. Conclusiones

«««< HEAD

Sabemos que, por más eficientes que sean los sistemas criptográficos que usamos actualmente, el constante aumento del poder computacional de los equipos y la demanda de privacidad en bloques de información cada vez más grandes nos hacen considerar la búsqueda de nuevos estándares y fundamentos teóricos que nos permitan prevenir futuros ataques. En ese sentido, el uso de pruebas de conocimiento cero en la criptografía actual podría significar no solo el mejoramiento de sistemas criptográficos conocidos y ampliamente usados, tal es el caso de RSA y los sistemas de llave pública en general, sino también de otorgar privacidad en donde la información de los usuarios se viera vulnerable a ser robada o falsificada, como en las cadenas de bloques. ======= Estrictamente hablando, el concepto de pruebas de conocimiento cero difiere de las pruebas formales o demostraciones. En lugar de presentar resultados obtenidos de demostraciones hechas de antemano para sustentarse, son más parecidas al proceso humano de convencimiento y más aún, el demostrador involucra al verificador en el esfuerzo por convencerlo de una proposición declarada como verdadera, manteniendo un corte interactivo y dinámico que contrasta con la demostración tradicional pasiva y estática. »»»> 3c4275024f911b3b6069649764781edc64849e17

Referencias

[1] Arora, S., Barak, B. *Computational complexity: a Modern Approach.* Beijing: World Publishing Corporation. 2012.

[2] Goldwasser, S., Micali, S., Rackoff, C. *The knowledge complexity of interactive proof-systems.* Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC 85. doi:10.1145/22145.22178. 1985.

[3] Quisquater, Jean-Jacques; Guillou, Louis C.; Berson, Thomas A. *How to Explain Zero-Knowledge Protocols to Your Children.* Advances in Cryptology - CRYPTO '89: Proceedings 435: 628-631. 1990.

[4] Peikert C, *Proofs of Knowledge* https://wiki.cc.gatech.edu/theory/images/5/54/Lec
Theoretical Foundations of Cryptography. Georgia Tech. 2010