SEGURIDAD DE LA INFORMACIÓN

TEMA 3

ESQUEMAS, PROTOCOLOS Y MECANISMOS DE SOPORTE
(A LA SEGURIDAD DE APLICACIONES Y DE REDES)

MATERIAL ADICIONAL, NO EVALUABLE

Zero-Knowledge Proof / Prueba de Conocimiento Cero

- Una de las partes (*probador*) proporciona la prueba de la veracidad de una declaración al *verificador*, sin revelar más que la veracidad de la declaración
  - **Propiedad**: Poseo una clave privada determinada, sin revelarla
  - **Balance**: Mi cuenta tiene suficiente dinero para realizar esta operación, sin revelar el balance
  - **Afiliación**: Pertenezco a un grupo, sin mostrar mi identidad
  - **Búsqueda del tesoro**: Conozco la localización del tesoro, pero no revelo su localización
  - **Sudoku**: Conozco la solución a un rompecabezas, pero sin revelarla
  - **Subasta**: Probar quien ha ganado, sin revelar las pujas

Zero-Knowledge Proof / Prueba de Conocimiento Cero

- Inicialmente propuesto en 1987 por Shafi Goldwasser, Silvio Micali and Charles Rackoff
  - No era práctico: requería de múltiples interacciones, y de pruebas de longitud inviable
- Todo cambio a partir de 2007, con el uso de primitivas criptográficas novedosas como los grupos bilineares / "pairings", curvas elípticas, etc
  - Groth and Sahai (2007)
  - Gennaro et al. (2012)
  - Eli Ben-Sasson et al. (2013)
- Actualmente, el protocolo más utilizado se conoce como **zk-SNARKs**

zk-SNARKs – Ventajas y Características

- *Zero-Knowledge Succint Non-Interactive Argument of knowledge*
  - *"Zero-Knowledge"*: El probador convence al verificador de la veracidad de un hecho, sin revelar más información
  - *"Non-Interactive"*: El probador no necesita interactuar con el verificador, la existencia de la prueba es suficiente
  - *"Succint"*: Permite realizar verificaciones en 5 ms, con pruebas de longitud 288 bytes
- Existen también propiedades relacionadas con los zk-SNARKs
  - Totalidad: El verificador quedará convencido (con una probabilidad abrumadora) que el hecho es veraz
  - Solvencia: Un hecho falso tiene una probabilidad ínfima de probarse

zk-SNARKs – Inconvenientes

- Actualmente, existen los siguientes inconvenientes:
  - "Trusted setup": La inicialización debe realizarse usando un protocolo seguro multiparte, y destruyendo cierto material que podría romper todo el protocolo en caso de que se hiciera público
  - Generación de las pruebas: Actualmente, es necesario de 5 a 40 segundos para generar una prueba (5-40ms para verificarla)
  - Tamaño: Los parámetros públicos (para la generación de las pruebas) pueden ser muy grandes, del orden de 750MB-1GB.

zk-SNARKs – Funcionamiento

- Desde un punto de vista abstracto, un zk-SNARK funciona de la siguiente forma:
  - OBJETIVO: Probar que el probador conoce un valor w, de tal forma que `C(x,w) = verdadero`, siendo x público
    - Ejemplo: Sin revelar w, probar que sha256(w) == x [C(x, w)]
  - Funciones:
    - G(λ, C) = (pk, vk), siendo C el programa, λ un parámetro secreto, pk la clave de prueba, y vk la clave de verificación
    - P(pk, x, w) = π, siendo π la prueba del probador
    - V(vk, x, π) = verdadero, si y sólo si π es correcto

Campos de aplicación

- Actualmente, los protocolos zk-SNARKs se están estudiando dentro de diversos esquemas “blockchain”, como Ethereum y zCash
  - *Ethereum*: Un algoritmo general de verificación se ejecuta dentro de un “smart contract”, usando los parámetros específicos (vk, x, π). La salida de la verificación puede propiciar la ejecución de más elementos del “smart contract”
    - Anonimato: emisor $\Rightarrow$ receptor: hash del balance, sabiendo que el balance del emisor es mayor que la cantidad enviada
  - zCash: “Shielded transactions”
    - Toda la transacción es anónima, y la(s) prueba(s) demuestran todas las condiciones necesarias para una transacción en el blockchain (balances, claves, etc)