Caso de uso en logística para el protocolo criptográfico "Zero Knowledge Proof"

Ciro Edgardo Romero[1] , Juan Augusto Pose[1]

[1] Laboratorio de Investigación, Desarrollo e Innovación
idi@cys.com.ar,
[2] C & S Informática S.A

Resumen. En diversos sistemas de seguimiento logístico se utilizan dispositivos de monitoreo y seguimiento. En estos casos, es normal emplear vehículos equipados con GPS o aplicaciones móviles para transmitir información relevante. Este esquema operativo da por sentado que todos los actores involucrados tienen acceso tanto a detalles sensibles sobre los bienes transportados como a sus especificaciones: su origen, su emisor y destinatario, entre otros datos.
Sin embargo, esta definición expone datos críticos a cualquier agente externo capaz de interactuar con el sistema. Esto provoca que, en presencia de actores malintencionados, surjan vulnerabilidades significativas en términos de seguridad y privacidad, comprometiendo la integridad del proceso de envío. Este trabajo explora la oportunidad de integrar técnicas criptográficas avanzadas, específicamente pruebas de Conocimiento Cero (ZKPs), para mejorar la seguridad y privacidad en logísticas, mientas se cumplen los requisitos operativos.

Palabras clave: Blockchain, Internet de las cosas, Protocolo ZK

Logistics use case for the "Zero Knowledge Proof" cryptographic protocol

Abstract. In various logistical tracking systems, monitoring and tracking devices are employed. Typically, vehicles equipped with GPS or mobile applications are used to transmit relevant information. Such operational schemes assume that all involved parties have access not only to sensitive details about the transported goods but also to their specifications, including: origin, issuer, recipient, and other pertinent data.
However, this approach exposes critical information to any external agent capable of interacting with the system. Consequently, in the presence of malicious actors, significant vulnerabilities arise concerning security and privacy, thereby compromising the integrity of the shipping process. This study explores the potential integration of advanced cryptographic techniques, specifically Zero-Knowledge Proofs (ZKPs), to ensure security and privacy within logistic operations while satisfying operational requirements.

Keywords: Blockchain, Internet of things, Zero Knowledge Proof

1 Introducción

En el sentido clásico, el propósito de la trazabilidad es tener una supervisión activa del traslado de productos: desde un punto de origen hasta un destino final. La intención de esta supervisión es identificar el origen y estudiar las diferentes etapas a lo largo del proceso de distribución, hasta que lleguen al consumidor final, tal y como explica en Campos and Ramírez, 2008. El progreso de la tecnología IoT [3] ha ayudado a las empresas involucradas en actividades logísticas, a usar dispositivos inteligentes para rastrear productos. Estos dispositivos permiten recopilar, procesar y enviar datos relacionados con la transferencia de bienes a programas especializados y, por lo tanto, crean un sistema de seguimiento activo.

1.1 Problemática de la seguridad y la privacidad

Según lo interpretado en Cordoves Mustelier and Frutos, 2024, un sistema de logística informático, recopila información de los usuarios para funcionar adecuadamente. A partir de dicha información, se crean registros de emisores y destinatarios de mercadería; esta información es utilizada para auditar la trazabilidad en una ruta de traslados. Al mismo tiempo, es consulta por los diferentes actores involucrados en dicha ruta. Esto es posible, porque los sistemas cuentan con bases de datos, propias o de terceros, donde queda almacenada la información sensible. Este tipo de almacenamiento vulnera la privacidad de la información que los usuarios otorgan, con el único propósito de hacer funcionar al sistema. Al mismo tiempo, esta información es confiada a los mecanismos de seguridad y políticas de privacidad de las empresas. Este panorama deja a los usuarios en una incertidumbre acerca de quién tiene acceso a su información potencialmente sensible.

1.2 Caso de estudio y prototipado

El presente trabajo se basa en la experiencia obtenida durante el desarrollo de un prototipo capaz de reportar, en tiempo real, el estado de las mercaderías trasladadas. Este proyecto, desarrollado íntegramente en el laboratorio, se centra en el monitoreo de variables ambientales (tanto internas como externas) de un contenedor térmico inteligente (ver Romero, 2024).

Esta iniciativa responde a una problemática real en el transporte de mercancías, donde las condiciones ambientales pueden alterar el estado de productos biodegradables. Tal como se ha observado en estudios previos (ver Navarro, 2013), factores externos al contenedor podrían afectar negativamente o incluso dañar el contenido de este, si no son gestionados adecuadamente.

[3] de la sigla en ingles *Internet of Things*, que se describen como la red de objetos físicos que se conectan a internet y pueden intercambiar datos

Resumen del prototipo La lógica para enviar mediciones se construye a partir de un dispositivo que forma parte de un contenedor térmico. Este dispositivo, esta basado en un microcontrolador ESP32[4], obtiene lecturas de distintos sensores y las agrupa en una estructura con formato JSON. Esta se envía como una *request* a un servicio web tipo API REST (mas detalles en La Paz, 2015) que procesa y almacena la información. Al mismo tiempo, la información relacionada al viaje y a su estado final, la cual refleja si las mediciones fueron aprobadas o no, queda persistida en una base de datos.

Todos los elementos descritos, son gestionados en contenedores para facilitar la ejecución de todos los componentes, tal y como está explicado en el artículo de Romero, 2024.

2 Prueba de conocimiento cero

Basados en el experimento mencionado en la sección anterior, se planteó la posibilidad de integrar un mecanismo para preservar la confidencialidad de los datos. Datos como la identidad de los sensores o los valores exactos capturados. Este mecanismo debe aumentar la capacidad de verificar las condiciones requeridas para el transporte, sin vulnerar la confidencialidad de los datos. Bajo tales propósitos, se pudo estudiar en Quisquater et al., 1989, un protocolo conocido como "pruebas de conocimiento cero"; también llamadas *Zero Knowledge Proofs* (*ZKPs*). Este protocolo, es un mecanismo criptográfico que permite a determinados usuarios (llamados *prover*) demostrar que cierta información es válida, sin revelar los datos subyacentes, a otros usuarios (llamados *verifier*). Este principio es fundamental en sistemas donde la privacidad y la seguridad son prioritarias, ya que garantiza la integridad de la información sin comprometer su confidencialidad. En el contexto de este trabajo, las ZKPs se utilizan para validar las mediciones de los sensores durante el transporte de productos, para asegurar que los parámetros establecidos (por ejemplo rangos de temperatura, entre otros) fueron respetados pero sin exponerlos.

Una cualidad destacable de la tecnología estudiada, es que una vez generada la prueba, esta es inmutable y puede ser verificada por cualquier tercero autorizado. Esta característica, evita la falsificación o manipulación retroactiva. Incrementa la resiliencia del sistema frente a ataques informáticos.

2.1 Circuitos aritméticos

Según la interpretación básica de Cunningham, 1994, un circuito aritmético es una representación matemática de un conjunto de operaciones algebraicas que se realizan sobre entradas para producir salidas específicas. En el contexto de las pruebas criptográficas, un circuito aritmético modela las restricciones o condiciones que deben cumplirse, o no, para que una declaración sea verdadera (ver Delfs et al., 2002). En la figura 1 se muestra un circuito aritmético básico, donde se multiplica el valor de "X" e "Y" para luego sumar "Z" a su resultado.

[4] microcontrolador fabricado por Espressif Systems, comúnmente promocionado para desarrollos IoT

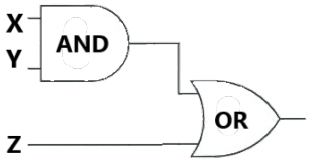

Fig. 1. Circuito aritmético básico.

Según las necesidades del experimento realizado, se requirió verificar que todas las mediciones de temperatura de un sensor estén dentro de un rango específico. El circuito aritmético define estas condiciones como operaciones algebraicas, las cuales se resuelven a través de entradas proporcionadas por el *prover*. A través de este proceso, se verifica si la declaración de entrada es válida.

2.2 Pruebas criptográficas: Zk-SNARKs

Los zk-SNARKs (*Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge*) son un tipo avanzado de prueba criptográfica dentro de la familia de las *Zero Knowledge Proofs*. Los zk-SNARKs permiten que un *prover* genere una prueba compacta de que posee cierto conocimiento, o que ha realizado un cálculo correcto. Lo anterior se logra sin revelar detalles adicionales sobre el conocimiento, o el cálculo en sí (mas detalles en Pinto, 2020). La tecnología SNARKs, responde a las siguiente características, que definen su nombre:

- Conciso o *Succinctness*: Las pruebas generadas son extremadamente compactas, típicamente condensadas en unos pocos kilobytes, independientemente del tamaño de los datos originales.
- Sin interacción o *Non-Interactive*: No requieren múltiples intercambios entre el *prover* y el *verifier*, lo que simplifica su implementación en sistemas distribuidos.
- Sin conocimiento o *Zero-Knowledge*: Garantizan que el *verifier* solo obtenga la certeza de que la declaración es verdadera o falsa, sin acceso a los datos subyacentes.

En el sistema propuesto, los zk-SNARKs se utilizan para generar pruebas criptográficas que validan las parámetros físicos, establecidos en el dispositivo. Estas pruebas son verificadas por el receptor o por una entidad logística, para intentar asegurar que las condiciones contractuales se cumplieron sin exponer los datos exactos registrados.

2.3 Lenguaje específico de dominio

Un lenguaje específico de dominio (DSL por sus siglas en inglés) es un lenguaje de programación diseñado para resolver problemas dentro de un dominio concreto,

en lugar de abordar tareas generales. A diferencia de los lenguajes de Propósito general (GPL, también por sus siglas en inglés), como Python o Java, los DSLs se enfocan en un ámbito particular, utilizando terminología, estructuras y reglas adaptadas a las necesidades de ese dominio. Ejemplos conocidos de este tipo de lenguajes son SQL, diseñado para gestionar bases de datos relacionales, o HTML, especializado en definir estructuras de páginas web.

El principal atractivo de los DSL, es su capacidad para simplificar la resolución de problemas. Al usar conceptos y sintaxis cercanos al dominio, permiten expresar soluciones de manera intuitiva y concisa, lo que facilita la colaboración entre especialistas del área y equipos técnicos al emplear terminología del dominio.

Noir es un DSL diseñado específicamente para el desarrollo de pruebas de conocimiento cero, un área crítica en criptografía y en sistemas de privacidad (ver Community, n.d.). Su objetivo es abstraer la complejidad matemática y algorítmica de las ZK proofs, permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones. A través de una sintaxis muy similar a Rust, es posible crear un programa que, una vez compilado, genere una abstracción conocida como ACIR (ver Cooper and Torczon, 2022). Este programa es capaz de ser transformado fácilmente en un circuito aritmético adaptable a varios sistemas de generación y verificación de pruebas criptográficas, como Halo2, PLONK, o Barretenberg. Estos últimos son utilizados en la industria por su eficiencia, flexibilidad y escalabilidad para generar y verificar pruebas de conocimiento cero. En la figura 2 se muestra un diagrama del funcionamiento básico del lenguaje Noir.

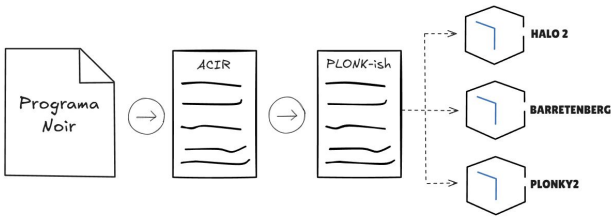

Fig. 2. Compilación y funcionamiento básico del lenguaje Noir.

Al mismo tiempo, Noir destaca por su interoperabilidad, ya que es compatible con diversas bibliotecas y herramientas criptográficas, que facilitan su integración en sistemas existentes, según todo lo estudiado en Lavin et al., 2024. Por otro lado, al tratarse de una tecnología emergente y en constante crecimiento, sus creadores junto con su comunidad, ofrecen soporte continuo, ademas de actualizaciones regulares.

3 Aplicación en la logística y la trazabilidad

Se utiliza el prototipo descrito en secciones anteriores como base para describir el siguiente experimento. Se aprovecha el sistema con sensores integrados en un contenedor, capaz de registrar métricas críticas (temperatura, humedad, etc.) durante un viaje. En una nueva funcionalidad, estos datos se procesan mediante un circuito definido en Noir, que evalúa umbrales predefinidos. El circuito junto con su verificador, actúan como una función booleana que evalúa las mediciones. En la figura 3 se muestra un diagrama con el sistema completo y sus diferentes componentes

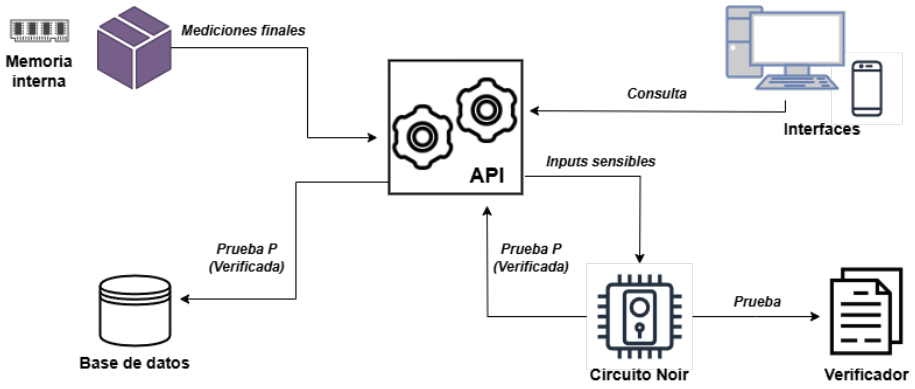

Fig. 3. Sistema completo con sus diferentes componentes.

3.1 Ejemplo práctico

Supongamos un escenario donde se considera el traslado del contenedor. Los requisitos para dicho traslado, refieren a que la temperatura del mismo, debe estar entre 10°C y 30°C. Estos datos son registrados por los sensores que forman parte del contenedor.

Prueba "P" Si todas las mediciones cumplen con los requisitos, el circuito genera una prueba criptográfica compacta, comúnmente llamada prueba ***"P"***. Esta prueba certifica que los datos son válidos, sin necesidad de revelar los valores exactos registrados.

Verificación Al finalizar el viaje, el *verifier* (una entidad logística o el receptor, por ejemplo) valida la prueba en tiempo real. Si la prueba es correcta, el estado del viaje se marca como "Mediciones aprobadas" ; de lo contrario, lo etiquetará bajo la denominación "Mediciones no aprobadas".

1. Caso exitoso: Si todas las mediciones están dentro del rango, el circuito genera la prueba ***"P"***, y el *verifier* confirma su validez.
2. Caso fallido: Si una medición excede 30°C, o cae por debajo de los 10°C, el circuito no puede generar la prueba ***"P"***, y el sistema automáticamente registra el incumplimiento.

3.2 Generación y verificación de pruebas

La generación de una prueba criptográfica, a partir de un circuito escrito en Noir, se logra por medio de una biblioteca Barretenberg desarrollada por Williamson and Andrews, n.d. Este proceso se compone de varios pasos provistos por la biblioteca a través su herramienta CLI. A continuación se muestra un circuito de ejemplo, con las especificaciones del rango de temperatura establecido.

```
// Circuito Noir, guardado con la extension .nr

fn main(input_tem: u8) {
    //Temperature check
    assert(input_tem >= tem.minValue,
     "Temperature below minimum value accepted");
    assert(input_tem <= tem.maxValue,
     "Temperature exceeeds maximum value accepted");
}
```

Pasos para generar la prueba criptográfica, a partir del circuito anterior:

1. En primera instancia, es necesario compilar el circuito por medio del comando **nargo compile**. Este proceso genera dos archivos: circuit.json (que representa el circuito en formato intermedio) y acir.gz (que contiene una versión comprimida del circuito en formato ACIR)
2. Seguidamente se debe validar que las entradas satisfacen las restricciones del circuito, lo cual se logra por medio de **nargo execute**. Si las restricciones no se cumplen, se mostrará un error indicando qué ecuaciones fallan.
3. La generación de la prueba, se realiza por medio del comando **bb prove**. Este comando produce un archivo **.proof**, el cual encapsula el cumplimiento de todas las restricciones del circuito.
4. Seguidamente la clave de verificación se realiza por medio del comando **bb write_vk**. El archivo .vk contiene los datos necesarios para cojear los datos, sin requerir acceso al circuito completo.
5. Finalmente, se utiliza el comando **bb verify** como verificación final. El mismo muetra un mensaje indicando si la prueba es válida, o no. La verificación es determinista y no requiere acceso a las entradas, logrando así la privacidad de los datos utilizados para generar la prueba.

4 Conclusiones y trabajo futuro

El mecanismo descrito en el presente trabajo, favorece que sólo los envíos con condiciones válidas reciban la aprobación, mitigando riesgos de fraude o errores

humanos. Basado en la problemática definida al principio del articulo, se pueden destacar otras características provechosas, tales como:

- La privacidad en los datos sensibles (por ejemplo, rutas, condiciones exactas), dado que no son expuestos durante el proceso de verificación.
- Integridad de las pruebas, ya que se garantiza que las mediciones no fueron alteradas y que cumplen con los requisitos contractuales.
- Escalabilidad independientemente de la cantidad de datos procesados, al tener pruebas zk-SNARK que cuentan con un tamaño fijo y pequeño (pocos kilobytes).
- Automatización en la verificación que se realiza (sin intervención humana) lo que reduce errores y costos operativos.

En otras palabras, la integración de ZKPs en el sistema de trazabilidad propuesto resuelve desafíos relativos a la logística y a la privacidad. Por otro lado, mejora la auditoría automatizada y fortalece el cumplimiento de normativas, respecto a parámetros cuantificados. Adicionalmente, al abstraer la complejidad técnica mediante herramientas como Noir, se facilita la adopción de criptografía avanzada en aplicaciones industriales.

Desde el equipo del Laboratorio de Investigación, Desarrollo e Innovación, se continuará el desarrollo para adaptarlo a las legislaciones vigentes en territorio argentino. Dado que los mecanismos de verificación de envíos también tienen que tener opciones de seguridad para evitar traslados peligrosos o malintencionados, se proyecta continuar las pruebas de campo en próximas etapas.

Paralelamente, será necesario ajustar la metodología de generación de pruebas que actualmente realizan los servicios, para que sean gestionadas directamente desde el dispositivo. Dicho proceso, está en etapa de investigación, con el objetivo de lograr la implementación de circuitos aritméticos codificados en el *firmware* del equipo. La finalidad última es alcanzar su funcionalidad autónoma.

Bibliografia

Campos, M. G. C., & Ramírez, C. S. (2008). *Análisis dinámico de sistemas industriales*. Trillas.

Community, N. (n.d.). Noir lang. https://noir-lang.org/docs

Cooper, K. D., & Torczon, L. (2022). *Engineering a compiler*. Morgan Kaufmann.

Cordoves Mustelier, D., & Frutos, M. (2024). Impacto y evolución de big data en la logística: Una revisión exhaustiva de tendencias y prácticas actuales. *Simposio Argentino de Informática Industrial e Investigación Operat (SIIIO 2024)-JAIIO 53 (Universidad Nacional del Sur, 12 al 16 de agosto de 2024)*.

Cunningham, D. R. (1994). *Basic circuit analysis*. John Wiley & Sons, Inc.

Delfs, H., Knebl, H., & Knebl, H. (2002). *Introduction to cryptography* (Vol. 2). Springer.

La Paz, R. L. G. (2015). *Desarrollo de aplicaciones web en el entorno servidor. ifcd0210*. IC Editorial.

Lavin, R., Liu, X., Mohanty, H., Norman, L., Zaarour, G., & Krishnamachari, B. (2024). A survey on the applications of zero-knowledge proofs. *arXiv preprint arXiv:2408.00243*.

Navarro, H. (2013). Logística en la cadena de frío. *ISBN 2950719007*, 34–37.

Pinto, A. M. (2020). An introduction to the use of zk-snarks in blockchains. *Mathematical Research for Blockchain Economy: 1st International Conference MARBLE 2019, Santorini, Greece*, 233–249.

Quisquater, J.-J., Quisquater, M., Quisquater, M., Quisquater, M., Guillou, L., Guillou, M. A., Guillou, G., Guillou, A., Guillou, G., & Guillou, S. (1989). How to explain zero-knowledge protocols to your children. *Conference on the Theory and Application of Cryptology*, 628–631.

Romero, C. E. (2024). Uso de un contenedor inteligente en la logística del traslado de mercadería. *4to Congreso Virtual de Microcontroladores y sus Aplicaciones*, (AJEA 40), 89–92. https://doi.org/10.33414/ajea.1757.2024

Williamson, Z., & Andrews, J. (n.d.). Aztec Labs — Building the Endgame for Blockchain Privacy. https://www.aztec-labs.com/