zk-SNARKs: Giao Thức Zero-Knowledge & Trusted Setup

Meta Description

zk-SNARKs là giao thức Zero-Knowledge Proofs (ZKP) phổ biến, ứng dụng trong Zcash. Phân tích R1CS, so sánh Groth16, PLONK, Marlin & rủi ro Trusted Setup!

Giới Thiệu

Trong hệ thống **Zero-Knowledge Proofs (ZKP)**, **zk-SNARKs** (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) là một trong những giao thức quan trọng nhất, giúp chứng minh một tuyên bố mà không tiết lộ bất kỳ thông tin nào.

zk-SNARKs được ứng dụng rộng rãi trong **blockchain, bảo mật giao dịch, và xác minh tính toàn vẹn của dữ liệu** mà không cần chia sẻ nội dung. Một số hệ thống nổi bật sử dụng zk-SNARKs bao gồm **Zcash, Filecoin, và Ethereum Layer 2**.

Trong bài viết này, chúng ta sẽ tìm hiểu:

- **zk-SNARKs là gì và cách hoạt động**
- **Cơ chế R1CS trong SNARKs**
- **So sánh Groth16, PLONK, Marlin – Ba thuật toán SNARKs phổ biến**
- **Trusted Setup là gì và những rủi ro liên quan**

Hãy cùng khám phá! 🚀

Key Takeaways

✅ **zk-SNARKs** là một giao thức ZKP không tương tác, giúp chứng minh thông tin mà không tiết lộ dữ liệu gốc.
✅ **R1CS (Rank-1 Constraint System)** là cách biểu diễn tính toán dưới dạng hệ ràng buộc bậc nhất, giúp chuyển đổi bài toán thành dạng SNARKs.
✅ **Groth16** có bằng chứng nhỏ và nhanh nhưng yêu cầu **Trusted Setup**. **PLONK** linh hoạt hơn, trong khi **Marlin** tối ưu cho các hệ thống phi tập trung.
✅ **Trusted Setup** là điểm yếu bảo mật của zk-SNARKs, vì nếu bị xâm phạm, có thể cho phép tạo bằng chứng giả.

Giới Thiệu Về zk-SNARKs

zk-SNARKs là một giao thức **Zero-Knowledge Proofs (ZKP)** không tương tác, nghĩa là người chứng minh (Prover) có thể tạo một bằng chứng duy nhất để người kiểm tra (Verifier) xác minh mà không cần trao đổi thông tin thêm.

Ba đặc điểm chính của zk-SNARKs

✔ **Zero-Knowledge**: Người kiểm tra không học được gì ngoài sự thật của tuyên bố.
✔ **Succinct**: Bằng chứng có kích thước nhỏ và thời gian kiểm tra nhanh.
✔ **Non-Interactive**: Chỉ cần một thông điệp duy nhất, không yêu cầu tương tác.

Ứng dụng phổ biến của zk-SNARKs

🚀 **Zcash** – Bảo mật giao dịch mà không tiết lộ số tiền hoặc địa chỉ.
🚀 **Ethereum Layer 2** – Giúp mở rộng quy mô bằng **zk-Rollups**.
🚀 **Filecoin** – Xác minh lưu trữ dữ liệu mà không cần tiết lộ nội dung.

💡 **Điểm đặc biệt:** zk-SNARKs nhanh hơn so với các giao thức ZKP khác như zk-STARKs, nhưng yêu cầu **Trusted Setup**, một điểm yếu bảo mật quan trọng.

Cơ Chế zk-SNARKs: R1CS (Rank-1 Constraint System)

Một trong những bước quan trọng trong zk-SNARKs là **chuyển đổi bài toán tính toán thành hệ phương trình đại số**. R1CS **(Rank-1 Constraint System)** là cách tiếp cận phổ biến nhất để làm điều này.

R1CS Hoạt Động Như Thế Nào?

1 Chuyển đổi chương trình thành mạch số học

- Biểu diễn các phép tính cộng, nhân bằng các cổng logic.

2 Chuyển đổi thành hệ ràng buộc R1CS

- **Cổng nhân**: Nếu z = x * y, ta có ràng buộc (x)*(y) - z = 0, với a chọn x, b chọn y, c chọn z.
- **Cổng cộng**: Nếu z = x + y, ta biểu diễn z - x - y = 0, với a chọn z, b chọn 1 (biến "one" luôn bằng 1), c chọn x + y, như trên Understanding Polynomial Commitments.

3 Chuyển đổi R1CS thành Quadratic Arithmetic Program (QAP)

- QAP là dạng mà SNARKs có thể sử dụng để tạo bằng chứng.

Ví dụ:
Giả sử cần tính y = x^2 + x + 1, ta có:

- temp1 = x * x, ràng buộc: x * x = temp1
- temp2 = x + 1, ràng buộc: temp2 - x - 1 = 0 (sử dụng biến "one" = 1)
- y = temp1 + temp2, ràng buộc: y - temp1 - temp2 = 0

💡 **R1CS giúp chuyển đổi các bài toán phức tạp thành hệ phương trình dễ dàng xử lý bằng zk-SNARKs.**

So Sánh Groth16, PLONK, Marlin – Ba Thuật Toán zk-SNARKs Phổ Biến

Thuật toán	Kích thước bằng chứng	Thời gian kiểm tra	Yêu cầu Trusted Setup	Linh hoạt
Groth16	Nhỏ (~96 bytes)	Nhanh	✅ Có	❌ Thấp
PLONK	Trung bình (~384-480 bytes)	Trung bình	⚠ Có hoặc không	✅ Cao
Marlin	Nhỏ (~96 bytes)	Nhanh	✅ Phi tập trung	✅ Trung bình

📌 Groth16 – Hiệu suất cao, nhưng cần **Trusted Setup**, ít linh hoạt.
📌 PLONK – Linh hoạt, không cần setup lại khi thay đổi bài toán, nhưng bằng chứng lớn hơn.
📌 Marlin – Giống Groth16 nhưng có setup phi tập trung, giảm rủi ro bảo mật.

Trusted Setup & Những Rủi Ro Bảo Mật

🔹 **Trusted Setup là gì?**
Trusted Setup là giai đoạn tạo tham số khởi tạo cho zk-SNARKs. Nếu các tham số này bị xâm phạm, có thể dẫn đến bằng chứng giả.

🔹 **Lý do cần Trusted Setup**

- Đảm bảo tính toán trên cặp ghép elliptic curve.
- Tăng hiệu suất bằng cách sử dụng **Common Reference String (CRS)**.

🔹 **Lỗ hổng bảo mật**
❌ **Nếu khóa bí mật bị lộ**, kẻ tấn công có thể tạo bằng chứng giả mà vẫn được hệ thống chấp nhận.
❌ **Cần tin tưởng vào quy trình setup**, nếu một bên độc hại tham gia, có thể phá vỡ bảo mật.

🔹 **Giảm rủi ro bằng Multi-Party Computation (MPC)**

- Nhiều người tham gia tạo khóa ngẫu nhiên.
- Không ai biết toàn bộ khóa bí mật.

- Nhưng vẫn có rủi ro nếu một số người thông đồng.

💡 **Giải pháp thay thế?** zk-STARKs không cần Trusted Setup, nhưng bằng chứng lớn hơn.

Kết Luận

✅ **zk-SNARKs** là một công nghệ mạnh mẽ giúp xác minh thông tin mà không tiết lộ dữ liệu.
✅ **R1CS** giúp chuyển đổi tính toán thành hệ ràng buộc đại số để SNARKs có thể xử lý.
✅ **Groth16, PLONK, Marlin** có ưu nhược điểm khác nhau, với Groth16 phổ biến nhưng yêu cầu Trusted Setup.
✅ **Trusted Setup** là điểm yếu bảo mật quan trọng, có thể bị xâm phạm nếu không thực hiện đúng.

💡 **Bài tiếp theo:** zk-STARKs - Công Nghệ Zero-Knowledge Proof Không Cần Trusted Setup