**2. Động lực phát triển**

- Nói về vấn đề các phương pháp fuzz testing hiện tại gặp khó khăn trong việc phát hiện lỗ hổng giữa các hợp đồng thông minh giao tiếp chéo

- Đưa ra ví dụ rằng sFuzz và Confuzzius không bao phủ được một số hàm do các vấn đề trong việc xử lý địa chỉ mặc định và các hợp đồng phụ. Còn xFuzz sử dụng mô hình học máy để giảm số lượng chuỗi giao dịch cần thử, nhưng vẫn bỏ qua một số hàm

=> CrossFuzz giúp giải quyết các thách thức đã được đưa ra bằng cách tập trung vào các chiến lược tạo tham số của hàm tạo và đột biến chuỗi giao dịch để cải thiện khả năng bao phủ mã và phát hiện lỗ hổng

**3. Background**

**3.1. Hợp đồng thông minh Ethereum**

- Hợp đồng thông minh Ethereum là các chương trình máy tính chạy trên nền tảng chuỗi khối Ethereum. Solidity là ngôn ngữ lập trình được sử dụng để viết các hợp đồng thông minh Ethereum, được biên dịch thành mã byte bởi các trình biên dịch như solc và được thực thi bởi Máy ảo Ethereum (EVM). Cụ thể, hợp đồng thông minh Solidity bao gồm nhiều hợp đồng, mỗi hợp đồng bao gồm một số biến trạng thái và chức năng. Các hàm được thực thi thông qua các giao dịch, bao gồm người gửi, chữ ký hàm, giá trị tham số hàm và lượng ether mang theo

- Trong paper thì f sẽ đại diện cho các giao dịch, p đại diện cho giá trị của các tham số trong hàm

**3.2. Kiểm thử Fuzzing và ConFuzzius**

**-** Kiểm thử fuzzing tạo các test case với giá trị ngẫu nhiên và gửi chúng để thực thi chương trình cần kiểm tra rồi dựa trên các điều kiện đường dẫn thu thập được, nó xác định xem có lỗ hổng bảo mật hoặc hành vi bất thường nào bị kích hoạt không. Sau đó, các test case con được tạo ra theo thuật toán di truyền và tiếp tục được gửi để kiểm tra chương trình, cho đến khi đạt điều kiện dừng (ví dụ: thời gian kiểm thử đã định)

- Giới thiệu về ConFuzzius ( Theo ChatGPT dịch: ConFuzzius kết hợp sáng tạo các kỹ thuật giải quyết ràng buộc (constraint-solving) với phương pháp fuzz testing để giải quyết vấn đề "magic byte" trong fuzz testing, nơi các nhánh có điều kiện nghiêm ngặt khó được bao phủ bằng các đầu vào ngẫu nhiên, làm giảm khả năng bao phủ mã của quá trình fuzz testing. ConFuzzius cũng xem xét sự phụ thuộc Read-After-Write (RAW) giữa các biến trạng thái, cải thiện hàm fitness để tối ưu chiến lược chọn seed, và chỉ kết nối các chuỗi giao dịch thỏa mãn điều kiện RAW trong giai đoạn lai ghép chuỗi giao dịch. Sau đó, chuỗi giao dịch được chia thành hai chuỗi mới để tạo test case con )

- Nêu ra 2 hạn chế của ConFuzzius:

+ Không hỗ trợ phát hiện lỗ hổng trong các tình huống hợp đồng giao tiếp chéo.

+ Sử dụng các giá trị mặc định cho tham số của hàm khởi tạo khi triển khai hợp đồng, bỏ qua ảnh hưởng của các cuộc gọi chéo giữa các hợp đồng đối với dòng dữ liệu

=> CrossFuzz ra đời là phương pháp mở rộng của ConFuzzius