**CS251: Cryptocurrencies and Blockchain Technologies**

Assignment #1

**Problem 1. A broken proof of work hash function**: Chúng ta đã thảo luận về việc sử dụng một hàm băm cho cơ chế *proof-of-work*. Một khi một và difficulty level được đưa ra, thì ta cần phải tìm một sao cho . Giả sử với bất kỳ (ở đây cho ) và xét hàm băm sau:

với

Chứng minh rằng hàm băm là không an toàn cho cơ chế *proof-of-work*. Giả sử là cố định, do đó cần chứng minh một kẻ tấn công thông minh có thể tìm được solution mà mất ít công sức nhất mỗi khi một được đưa ra.

***Hint***: Kẻ tấn công làm hầu hết công việc trước khi được phát hành.

**Problem 2. Beyond binary Merkle tree**: có thể sử dụng cây () nhằm một danh sách các phần tử để sau này cô ấy có thể chứng minh với Bob rằng , bằng cách sử dụng một chứa nhiều nhất các giá trị băm. Trong problem này, mục tiêu của mình là giải thích được bằng cách sử dụng (đây là một biến thể của cây với mỗi nút không phải là nút lá thì có tối đa nút con). Giá trị băm của tất cả các nút không phải là nút lá được tính bằng cách băm một chuỗi ghép của các giá trị nút con của nút đấy.

1. Giả sử . Giải thích làm cách nào mà tính được bằng cách dùng (tức là với ) . Rồi sau đấy làm thế nào để chứng minh nằm trong ? Những giá trị nào là cần thiết để cung cấp trong .
2. Giả sử chứa phần tử. Độ dài của mà chứng minh là như thế nào, biểu diễn dưới dạng hàm của và .
3. Với lớn, nếu chúng ta muốn tối ưu kích thước của , sẽ là tốt hơn khi dùng hay ? Vì sao?

**Problem 3. Bitcoin script**: đang trong một chuyến du lịch và cô ấy lo rằng thiết bị lưu trữ bị đánh cắp. Cô ấy muốn lưu bằng cách nào đó mà người ta có thể đổi được bằng mật khẩu . Theo đó, lưu theo address như sau:

1. Viết mã sẽ đổi thành công này với mật khẩu .

***Hint***: Mã chỉ cần dài một dòng.

1. Giả sử chọn mật khẩu dài sáu ký tự. Giải thích tại sao số của cô ấy có thể bị đánh cắp khi của cổ được post lên blockchain. Chúng ta có thể giả định rằng việc tính băm một chuỗi sáu ký tự với có thể thực hiện trong thời gian hợp lý.
2. Giả sử chọn một cụm mật khẩu dài ký tự (đủ mạnh). Liệu trên có đủ an toàn để bảo vệ của ? Tại sao?

***Hint***: Giải thích thông qua những gì xảy ra khi mà cố gắng đổi lấy của mình.

**Problem 4. BitcoinLotto**: Giả sử quốc gia Bitcoinia quyết định chuyển đổi xổ số quốc gia của mình sang Bitcoin. Một nhà máy in vé cào đáng tin cậy tồn tại và sẽ không lưu giữ hồ sợ về bất kỳ giá trị nào được in. Bitcoinia đề xuất một thiết kế đơn giản như sau: một public address hàng tuần chứa jackpot được phát hành. Điều này cho phép mọi người có thể chắc chắn là jackpot tồn tại, tức tồn tại một UTXO liên kết với public address chứa jackpot. Sau đó các ticket được in ra sao cho vé độc đắc là chứa private key tương ứng.

1. Nếu như người thắng cuộc tìm được vé và thứ Hai và ngay sau đó thông báo rằng là mình thắng cuộc, thì điều này là không tốt về mặt sale vì những người chơi khác sẽ nhận ra giải thưởng đã có người thắng cuộc. Chỉnh sửa một chút trong thiết kế ban đầu, đó là sử dụng một trong các locktime opcode để đảm bảo rằng giải thưởng chỉ được thông báo vào cuối tuần (tất nhiên là chúng ta không ngăn được việc người thắng cuộc chứng minh quyền sở hữu của private key). Bitcoin script opcodes được liệt kê trong <https://en.bitcoin.it/wiki/Script> . Ngoài ra, hãy giải thích cách sử dụng multisig cho trường hợp này.
2. Một số vé sẽ được hủy hoặc mất đi. Chỉnh sửa một chút thiết kế trò chơi để cho những vé này không phải là jackpot của Tuần thì có thể là jackpot của Tuần . Nếu Tuần không phải là jackpot thì những vé từ cả Tuần và đều có thể là jackpot của Tuần n + 2, và tiếp tục. Hãy đề xuất một phương án để không phải bắt người quản trị không tham ô tiền vượt quá khả năng có thể dựa trên thiết kế ban đầu ? Chúng ta có thể giả định hệ thống xổ số chạy trong 1000 tuần và sau đó tắt vĩnh viễn. ***Hint***: Dùng multisig.

**Problem 5. Lightweight clients**: Giả sử Bob đang chạy một ứng dụng Bitcoin client siêu nhẹ, ứng dụng nhận header hiện tại của blockchain từ nguồn tin cậy. Bitcoin client này hạn chế về bộ nhớ nên nó chỉ lưu trữ các block header của các block gần đây nhất trong blockchain, xóa những block trước đó.

1. Xét trong thiết kế Blockchain hiện tại, mỗi block chứa root của Merkle tree của tất cả các transaction trong block và băm của block header liền trước. Giả sử Alice post một transaction vào chuỗi mà transaction này gửi tiền cho Bob. Sau này, trong một số lần Alice muốn chứng minh rằng cô ấy đã gửi tiền cho Bob. Bob chạy ứng dụng Bitcoin client lên và chỉ có block header gần nhất. Vậy những thông tin nào Alice nên gửi cho Bob để chứng minh transaction trên đã nằm trong một block nào đó trên Blockchain?
2. Giả sử payment của Alice nằm trong block thứ k tính từ block gần nhất và mỗi block có đúng n transaction. Ước lượng cần bao nhiêu byte proof tính theo hàm của n và k. Tính kích thước của proof với và là hàm băm được sử dụng.
3. Một đề xuất đó là thêm một trường trong tất cả block header mà chứa băm của block header trước đó. Ví dụ, block header h chứa băm của block header h-1 và băm của block header m < h, với m được tính bằng cách: Gọi là lũy thừa lớn nhất của chia , thì . Minh họa bởi hình dưới. Giải thích làm thế nào để có thể giảm đi kích thước của proof ở câu b. Trường hợp xấu nhất thì kích thước proof là bao nhiêu, biểu diễn bằng hàm của và .

***Hint***: Giả sử block hiện tại của Blockchain là block , thì trường hợp xấu nhất cho kích thước proof là khi và Alice cố gắng chứng minh sự tồn tại của một transaction trong block 1.

