FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Teoria de la Programación

Informe Individual0

Etheroll

2do Cuatrimestre 2018

Nombre: Alejo Ivan Acevedo

Padron: 99146

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Indice:

Introducción

Como ya se dijo en el trabajo grupal, Solidity es un lenguaje que corre sobre Ethereum la cual es una red descentralizada basada en blockchain, que como ya vimos también, nació como una idea de libro contable transparente y fácilmente auditable. A su vez, Ethereum inculco la idea de smart-contract, el cual es un contrato que se cumple automáticamente cuando se dan ciertas condiciones. Por estas razones, muchas de las aplicaciones descentralizadas (DApps) desarrolladas en Solidity, son aplicaciones que necesitan manejar plata de manera confiable y transparente. Una de estas aplicaciones es Etheroll, el cual es un juego de apuestas de “dados”.

¿Que es Etheroll?

Etheroll como dije anteriormente es un juego de apuestas, este es probabilisticamente confiable y descentralizado, se basa en acertar si el numero que saldrá en un dado de 100 caras sera menor al que vos elegiste. La ganancias de la apuesta serán proporcional a la posibilidad de ganar la apuesta, es decir, si elegís el numero 5 tu ganancia sera mucho mayor que sí elegís el numero 98 pero también las chances de ganara serán mucho menores, finalmente el margen de la casa de ganar es de 1%. El juego esta desarrollado como un contrato que esta corriendo en la red de Ethereum.
A su vez Etheroll tiene su propio criptomoneda, llamada DICE, la cual cumple los estándares ERC20. El 100% de las ganancias de Etheroll se divide entre las personas que poseen esta criptomoneda, proporcionalmente a la cantidad que tiene cada persona. Ademas de asegurar la obtención de ganancias, al ser dueño de una cantidad de DICE, te da el derecho a votar en las decisiones que se deban llevar a cabo.

Análisis del uso del lenguaje

La eleccion de usar Solidity para este desarrollo se puede ver a simple vista, en la idea de que las apuestas sean completamente transparentes. Al estar el código del juego escrito completamente en un smart-contract el cual esta corriendo en la blockchain de Ethereum nos provee una transparencia que ningún casino online tradicional podría proveer. El contrato puede ser visto por cualquier usuario de hecho esta exhibido en la pagina donde se realizan las apuestas, a su vez asegura una paga inmediata a los ganadores ya que cuando el contrato constata que se cumplieron las condiciones necesarias para que el jugador gane inmediatamente el mismo envía el dinero a los usuarios que ganaron. A su vez al estar el contrato a la vista de todo cualquiera puede constatar que el margen de la casa es del 1% como así también que la generación de los números del dado es completamente aleatoria y no puede ser manipulado por ningún factor.

Análisis del Smart-Contract

El Smart-Contract completo lo podemos encontrar en la siguiente dirección: https://etheroll.com/#/smart-contract sin embargo en este informe nos vamos a centrar en la forma que el mismo genera los números del dado y nos asegura que la obtención del mismo es probabilisticamente justo.
Para la generación de este numero el código utiliza oraclize.it quien nos permite obtener un numero random entero desde afuera de la blockchain. Este entero es proveído por una API de random.org. Luego el contrato lleva acabo una encriptacion sha3 para obtener el resultado final del dado. De esta manera el contrato se asegura que el resultado no pueda ser manipulado por mineros ni otros terceros.
De esta forma, cuando se recibe una apuesta validad por medio de la función playerRollDice(uint rollUnder), el Smart-Contract genera una query parcialmente encriptado a la api provista por random.org. Y así se asegura que la clave que se utiliza para realizar el query a la api permanece seguro pero el request y el rango quedan a la vista para que el usuario confirme que Etheroll es honesto.

Las siguientes son las lineas de Solidity necesarias para realizar el pedido del. Numero aleatorio:

bytes32 rngId = oraclize_query("nested", "[URL] ['json(https://api.random.org/json-rpc/1/invoke).result.random.data.0', '\\n{\"jsonrpc\":\"2.0\",\"method\":\"generateSignedIntegers\",\"params\":{\"apiKe y\":${[decrypt] BBdNQjoFRtO/Od/8NmPt+rdMjLiRAciRv+NxvI5vtiSSgcrUFT9vx636i0xetTG5Rqy1tWtnG4Uaw7GuVeb3HDYHoS2WXFYBpoK+XqXQIzcQwzgiyHjyGN+yl1ia581kX0fYb8FLOcFGIvRtx6aas8+mfXpGFvk=},\" n\":1,\"min\":1,\"max\":100,\"replacement\":true,\"base\":10${[identity] \"}\"},\"id\":1${[identity] \"}\"}']", gasForOraclize);

Como se puede ver el numero pedido es exactamente como se espera:

n\”:1,\”min\":1,\"max\":100,\"replacement\":true

Y la clave de la api se encuentra encriptada:

{\"apiKe y\":${[decrypt] BBdNQjoFRtO/Od/8NmPt+rdMjLiRAciRv+NxvI5vtiSSgcrUFT9vx636i0xetTG5Rqy1tWtnG4Uaw7GuVeb3HDYHoS2WXFYBpoK+XqXQIzcQwzgiyHjyGN+yl1ia581kX0fYb8FLOcFGIvRtx6aas8+mfXpGFvk=}

Una vez que la api responde, el contrato utiliza el siguiente código para generar un numero random entre 1 y 100 inclusive el cual esta idénticamente distribuido en todo el rango:

finalResult = uint(sha3(r, p)) % 100 + 1;

Siendo r el numero random obtenido desde random.org y p la dirección IPFS del PageSigner TLSNotary

Finalmente el contrato constata si el numero elegido por el usuario es menor al obtenido y en caso de ser así le pago inmediatamente al apostador la suma de dinero correcta.

Bibliografia

https://crowdfund.etheroll.com/etheroll-whitepaper.pdf

https://etheroll.com/#/smart-contract

https://questions.coincheckup.com/etheroll/what-is-etheroll-dice/

https://github.com/bokkypoobah/TokenTrader/wiki/DICE-%E2%80%90-Etheroll