UNIVERSITÀ DEGLI STUDI ROMA TRE

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica

Tesi Di Laurea

Progettazione, sviluppo e prove sperimentali di un PoC di FulgurHub in Typescript

Laureando

Federico Ginosa

Matricola 457026

Relatore

Alberto Paoluzzi

Correlatore

Federico Spini

Anno Accademico 2017-2018

Ad Ada Lovelace

Indice

Elenco delle figure

Introduzione

- 2008 paper di Sathosi Nakamoto e pubblicazione protocollo BitCoin
- Stato del network dieci anni dopo
- Problemi di scalabilità
- Soluzioni al problema della scalabilità
  - Algoritmo di consenso
  - Sharding
  - OffChain
- Lavoro di questa tesi
  - Studio e analisi della sicurezza di NOCUST
  - Design e implementazione di un IPC
  - Progettazione, sviluppo e prove sperimentali di Fulgur Hub

Capitolo 1

Background

In questo capitolo si vuole fare una panoramica delle tecniche di scalabilità off-chain della blockchain. In sezione 1.1 vengono presentati la blockchain e gli smart contract. In sezione 1.2 si introducono gli state channel. In sezione 1.3 viene presentato il design di un inextinguishable payment channel, una particolare tipologia di state channel. Infine in sezione 1.3 viene introdotto Fulgur Hub, il design di un hub che permette di scambiare valore mediante l'uso di payment channel tra più di due entità.

Blockchain e smart contract

State channel

Gli state channel permettono alle parti di modificare in modo sicuro porzioni della blockchain (e.g. uno smart contract). Queste modifiche avvengono mediante lo scambio di messaggi off-chain (e.g. una cartolina, un sms, una richiesta http). I messaggi scambiati descrivono un aggiornamento di stato,

come l'aggiornamento del bilancio di una parte o la prossima mossa di un giocatore di tris. Uno state channel ha dunque due stati, quello on-chain e quello off-chain. Le operazioni che modificano lo stato on-chain vengono dette *operazioni on-chain*; quelle che modificano lo stato off-chain vengono dette *operazioni off-chain*.

Inextinguishable payment channel

Apertura di un payment channel

Schema propose/accept

Transazioni off-chain

Schema detach/attach

Introduzione

Hot withdraw

Hot refill

Chiusura di un payment channel

Threat model

Double spending di un token

Token non speso

Gestione della free-option

Fulgur Hub

Obiettivi di progettazione

Pagamenti ibridi

Trustless

Non censurabile

Capitolo 2

Progettazione

Descrizione generale dell'architettura

??? Alice -> Bob: Authentication Request Bob –> Alice: Authentication Response

Alice -> Bob: Another authentication Request Alice <– Bob: another authentication Response **???** ### Lo smart contract ### Il client ### L'hub ## Apertura di un canale ## Pagamenti omogenei ### Transazioni OnChain-OnChain ### Transazioni OffChain-OffChain ## Pagamenti ibridi ### Transazioni OffChain-OnChain ### Transazioni OnChain-OffChain ## Riscossione di un pending token ## Chiusura di un canale ## Threat model ### B non invia la ricevuta di pagamento ad A ### Generazione di una miriade di token ### L'hub non permette di attaccare un token ### L'hub non permette di staccare un token ### Tentativo di pagamento con un token scaduto ### A si rifiuta di regolare un trasferimento nei confronti dell'hub ### Tentativo di ritirare un pending token già usato ### Mancanza di cooperazione nel ricevere un pagamento

Capitolo 3

Implementazione

Lo smartcontract EthereumSmartContract

Client

ClientPrivateCommands

ClientPublicCommand

LevelDBClientDatabase

ClientMonitorService

Hub

HubPrivateCommands

HubPublicCommands

RedisHubDatabase

HubMonitorService

Capitolo 4

Prove sperimentali

Benchmark server

Transazioni OffChain-OffChain seriali

Transazioni OffChain-OffChain concorrenti

Conclusioni e sviluppi futuri

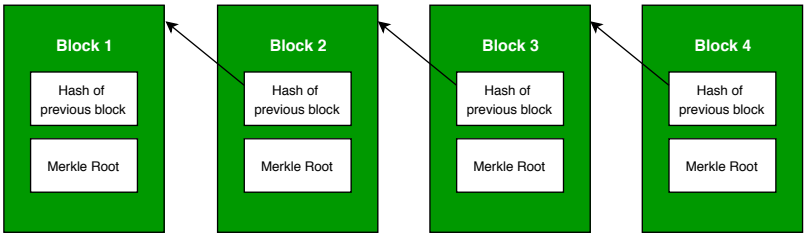

Figura 4.1: Blockchain con blocchi non manomessi