Tugas Minggu 1

Fajri Nurfauazan | 1103180184 | TK-42-PIL

Dalam makalah ini, penggusulan solusi untuk masalah pengeluaran ganda menggunakan server cap waktu terdistribusi peer-to-peer untuk menghasilkan bukti komputasi dari urutan kronologis transaksi. Sistem ini aman selama node yang jujur secara kolektif mengontrol lebih banyak daya CPU daripada kelompok node penyerang yang bekerja sama, Setiap pemilik mentransfer koin ke yang berikutnya dengan menandatangani secara digital hash dari transaksi sebelumnya dan kunci publik dari pemilik berikutnya dan menambahkan ini ke akhir koin, Setelah itu penerima pembayaran dapat memverifikasi tanda tangan untuk memverifikasi rantai kepemilikan.

Masalahnya tentu saja adalah penerima pembayaran tidak dapat memverifikasi bahwa salah satu pemilik tidak membelanjakan koinnya dua kali lipat, Solusi umum adalah memperkenalkan otoritas pusat tepercaya, atau mint, yang memeriksa setiap transaksi untuk pengeluaran ganda. Setelah setiap transaksi, koin harus dikembalikan ke mint untuk mengeluarkan koin baru, dan hanya koin yang dikeluarkan langsung dari mint yang dipercaya untuk tidak dibelanjakan ganda, Dan membutuhkan cara agar penerima pembayaran mengetahui bahwa pemilik sebelumnya tidak menandatangani transaksi sebelumnya

Untuk tujuan makalah ini, transaksi paling awal adalah yang diperhitungkan, jadi tidak peduli dengan upaya selanjutnya untuk membelanjakan dua kali lipat. Satu-satunya cara untuk mengonfirmasi tidak adanya transaksi adalah dengan mengetahui semua transaksi. Dalam model berbasis mint, mint mengetahui semua transaksi dan memutuskan mana yang datang lebih dulu. Penerima pembayaran perlu bukti bahwa pada saat setiap transaksi, mayoritas node setuju bahwa itu adalah yang pertama diterima, Server timestamp bekerja dengan mengambil hash dari blok item untuk dicap waktu dan mempublikasikan hash secara luas, seperti di surat kabar atau pos Usenet. Stempel waktu membuktikan bahwa data harus ada pada saat itu, jelas, untuk masuk ke hash

Setiap stempel waktu menyertakan stempel waktu sebelumnya dalam hashnya, membentuk rantai, dengan setiap stempel waktu tambahan memperkuat yang sebelumnya contohnya :

Pemilik Transaksi Hash 1 Pemilik Kunci Publik Tanda Tangan 0 Pemilik Transaksi Hash 2 Pemilik Kunci Publik 1 Tanda Tangan Hash Verifikasi Pemilik Transaksi 3 Pemilik Kunci Publik 2 Tanda Tangan Hash Verifikasi Pemilik 2 Pemilik Kunci Pribadi 1 Kunci Pribadi Sign Sign Pemilik 3 Kunci Pribadi 4. Bukti- Pekerjaan Untuk mengimplementasikan server timestamp terdistribusi secara peer-to-peer, kita perlu menggunakan sistem proofof-work yang mirip dengan Hashcash Adam Back, daripada posting koran atau Usenet.

Proof-of-work melibatkan pemindaian nilai yang ketika di-hash, seperti dengan SHA-256, hash dimulai dengan sejumlah bit nol. Untuk jaringan timestamp yang menggunakan pengimplementasian proof-of-work dengan menambahkan nonce di blok sampai ditemukan nilai yang memberikan hash blok nol bit yang diperlukan. Setelah upaya CPU telah dikeluarkan untuk membuatnya memenuhi bukti kerja, blok tidak dapat diubah tanpa mengulang pekerjaan. Karena blok selanjutnya dirantai setelahnya, pekerjaan untuk mengubah blok akan mencakup mengulang semua blok setelahnya

Maka bukti kerja juga memecahkan masalah penentuan representasi dalam pengambilan keputusan mayoritas, Keputusan mayoritas diwakili oleh rantai terpanjang, yang memiliki upaya bukti kerja terbesar yang diinvestasikan di dalamnya, Jika sebagian besar daya CPU dikendalikan oleh node yang jujur, rantai yang jujur akan tumbuh paling cepat dan melampaui rantai pesaing.

Untuk memodifikasi blok sebelumnya, penyerang harus mengulang proof-of-work blok dan semua blok setelahnya dan kemudian mengejar dan melampaui pekerjaan node yang jujur. Untuk mengimbangi peningkatan kecepatan perangkat keras dan minat yang bervariasi dalam menjalankan node dari waktu ke waktu, kesulitan proof-of-work ditentukan oleh rata-rata bergerak yang menargetkan jumlah rata-rata blok per jam. Node selalu menganggap rantai terpanjang sebagai yang benar dan akan terus bekerja untuk memperpanjangnya. Jika dua node menyiarkan versi berbeda dari blok berikutnya secara bersamaan, beberapa node mungkin menerima satu atau yang lain terlebih dahulu, Siaran transaksi baru tidak harus menjangkau semua node maka selama mereka mencapai banyak node, mereka akan masuk ke blok tak lama

Blokir siaran juga toleran terhadap pesan yang dijatuhkan, Jika sebuah node tidak menerima sebuah blok, ia akan memintanya ketika menerima blok berikutnya dan menyadari bahwa ia melewatkan satu blok. Ini akan menambah insentif bagi node untuk mendukung jaringan, dan menyediakan cara untuk mendistribusikan koin ke dalam sirkulasi, karena tidak ada otoritas pusat untuk menerbitkannya

Insentif juga dapat didanai dengan biaya transaksi, Jika nilai keluaran suatu transaksi lebih kecil dari nilai masukannya, selisihnya adalah biaya transaksi yang ditambahkan ke nilai insentif blok yang berisi transaksi tersebut. Setelah jumlah koin yang telah ditentukan telah memasuki sirkulasi, insentif dapat beralih sepenuhnya ke biaya transaksi dan sepenuhnya bebas inflasi. Insentif dapat membantu mendorong node untuk tetap jujur