Single use seal

Satoshi Spritz

Valerio Vaccaro

January 3, 2021

Table of contents

Cos'è?

Con single use seal indichiamo la possibilità di legare l'aggiornamento di un contenuto alla spesa di una UTXO.
Caratteristiche:

- funziona solo su blockchain transazionali,
- le informazioni saranno mantenute su mezzi separati dalla blockchain utilizzata,
- meglio se esiste un meccanismo per bloccare le UTXO,
- sulla blockchain vedo solo transazioni (eventualmente con opreturn),
- nativamente funzionante anche con logiche multisig.

A cosa serve?

Un single use seal può servire a:

- garantire un unico aggiornamento del contenuto,
- garantire privilegi sull'aggiornamento del contenuto solo all'avente diritto,
- garantire la storia e gli aggiornamenti di un contenuto,
- dare una data certa all'azione di aggiornamento del contenuto,
- favorire un pagamento atomico per l'aggiornamento del contenuto.

A cosa serve?

Un single use seal può servire a:

- garantire un unico aggiornamento del contenuto,
- garantire privilegi sull'aggiornamento del contenuto solo all'avente diritto,
- garantire la storia e gli aggiornamenti di un contenuto,
- dare una data certa all'azione di aggiornamento del contenuto,
- favorire un pagamento atomico per l'aggiornamento del contenuto.

Il contenuto può essere:

- lo stato di un file (e.g. stato di un portafoglio di investimenti),
- un certificato di proprietà su un oggetto,
- un token, ...

Esempio: generazione

Sono l'azienda ALFA e voglio creare una catena di single use seal.

- creo un documento con tutte le informazioni collegate al seal (DOCUMENTO ALFA),
- identifico una UTXO (UTXO ALFA) da collegare alla modifica del DOCUMENTO ALFA,
- riporto la UTXO ALFA nel DOCUMENTO ALFA,
- effettuo una transazione spendendo la UTXO BETA e mettendo tra gli output un committment di DOCUMENTO ALFA,
- memorizzo la transazione come transazione generatrice e il documento creato.

Esempio: aggiornamento

Sono l'azienda CHARLIE e voglio aggiornare il DOCUMENTO CHARLIE che ho ricevuto dallo step precedente passando la possibilità di aggiornare il documento all'azienda DELTA.

- chiedo all'azienda DELTA una UTXO che chiamerò UTXO DELTA,
- creo il DOCUMENTO DELTA che aggiorna il DOCUMENTO CHARLIE,
- aggiungo al DOCUMENTO DELTA la UTXO DELTA,
- spendo la UTXO CHARLIE (presente nel DOCUMENTO CHARLIE) e tra gli output aggiungo un commitment a DOCUMENTO DELTA,
- memorizzo il documento aggiornato.

Esempio: aggiornamento

Parallelamente l'azienda DELTA:

- fornisce la UTXO DELTA,
- controlla la storia del DOCUMENTO CHARLIE,
- controlla la correttezza del DOCUMENTO DELTA,
- controlla che al DOCUMENTO DELTA venga aggiunta la UTXO DELTA,
- contolla che la UTXO CHARLIE venga spesa e che ci sia il commitment desiderato,
- memorizza i documenti aggiornati.

Atomic swap

La transazione che spende la UTXO CHARLIE potrebbe altresi far ricevere all'azienda CHARLIE (da parte dell'azienda DELTA) i fondi collegati all'operazione.

Esempio: controllo

Sono l'azienda ECHO e voglio controllare che il DOCUMENTO DELTA sia l'ultimo disponibile ed aggiornato.

- chiedo la serie dei documenti dal generatore al documento in esame (DOCUMENTO DELTA),
- controllo che ci sia un path tra il generatore e delta (e che tutti gli hash siano corretti),
- controllo che la UTXO DELTA non sia stata spesa; se è così allora il DOCUMENTO DELTA è l'ultima versione disponibile del documento.

Eonpass

Eonpass è una soluzione capace di registrare tutti gli stati dei prodotti all'interno della filiera logistica dalla produzione ai servizi post vendita.

- ogni prodotto genera un gemello virtuale (e digitale) registrato in un generatore,
- ad ogni step del prodotto fisico corrisponde un aggiornamento del gemello virtuale,
- le modifiche al gemello virtuale sono registrate tramite la spesa di un seal,

Eonpass

- chi spedisce il bene è responsabile dell'aggiornamento delle informazioni sul gemello virtuale,
- chi riceve il bene è responsabile del controlle delle informazioni contenute nel gemello virtuale.

Risorse

https://eonpass.com/
https://gitlab.com/Eonpass/specs/blob/master/architecture.md

Liquidcert

Liquidcert usa i single use seal per:

- certificare la storia degli aggiornamenti di un documento,
- certificare la sequenza e l'ordine delle parti di un flusso di dati.

Risorse

https://liquidcert.it/
https://github.com/valerio-vaccaro/SeqChain

RGB

RGB usa single use seal per creare, distruggere o trasferire colored coin o altri asset.

Risorse

Da qualche parte su https://github.com/LNP-BP