Confidential Transactions
Theory Justification

Iftach Haitner*

July 23, 2025

Abstract

[**Iftach's Note: TODO**]

Contents

*Stellar Development Foundation. E-mail: `iftach.haitner@stellar.org`..

1 Introduction

[**Iftach's Note: TODO**]

2 Preliminaries

2.1 Notation

We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for integers and functions. Let $\mathbb{N}$ denote the set of natural numbers. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$ and $(n) := \{0, \ldots, n\}$. For a relation $\mathcal{R}$, let $\mathcal{L}(\mathcal{R})$ denote its underlying language, i.e., $\mathcal{L}(\mathcal{R}) := \{x \colon \exists w \colon (x, w) \in \mathcal{R}\}$.

2.2 Homomorphic Encryption

An homomorphic encryption is a triplet $(\mathsf{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ of efficient algorithms, with the standard correctness and semantic security properties. In addition, there is addition operation denote $+$ over any two (valid) ciphertexts such that for any validly generated public key pk and valid ciphertexts x_0, x_1, it holds that $\mathrm{Enc}_{sk}(x_0) + \mathrm{Enc}_{pk}(x_1) \in \mathrm{Supp}(\mathrm{Enc}_{sk}(x_0 + x_1 \bmod q)$, where $q \in \mathbb{N}$ is efficiently determined by pk.

[**Iftach's Note: Do we really need the homomorphic properties or only for the proofs?**]

3 The Confidential Transaction Protocols

3.1 The Ideal Functionality

Functionality 3.1 ($\mathcal{F}_{\mathsf{ConfTrans}}$: Confidential transactions)**.**

Parties: Issuer I, Chain C and users $\mathsf{U}_1, \ldots, \mathsf{U}_n$.

Init. Upon receiving init from all parties:

1. For each $i \in [n]$: $\mathsf{ActBalance}_i, \mathsf{PndBalance}_i \leftarrow 0$ and $\mathsf{log}_i \leftarrow \emptyset$.
2. $\mathsf{log} \leftarrow \emptyset$.

Issue. Upon receiving $(\mathsf{sid}, \mathsf{issue}, x, d)$ from C and I:

1. Assert($x \in \mathbb{N}$ and $d \in [n]$).
2. $\mathsf{PndBalance}_d \mathrel{+}= x$.
3. Set $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{issue}, x, d)$.

Transfer. Upon receiving $(\mathsf{sid}, \mathsf{transfer}, d)$ from C and U_s, with U_s using private input x.

1. Assert($x \in \mathbb{N}$, $\mathsf{ActBalance}_s \geq x$ and $s, d \in [n]$).
2. $\mathsf{ActBalance}_s \mathrel{-}= x$.

3. $\mathsf{PndBalance}_d \cup= x$.
4. Set $\mathsf{log}_d \cup= (\mathsf{sid}, \mathsf{transfer}, s, x)$
5. Set $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{transfer}, s, d)$

Rollover. Upon receiving $(\mathsf{sid}, \mathsf{rollover})$ from party U_i and C, party C

1. Set $\mathsf{ActBalance}_i \mathrel{+}= \mathsf{PndBalance}_i$.
2. Set $\mathsf{PndBalance}_i \leftarrow 0$.
3. Set $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{rollover}, i)$

Withraw. Upon receiving $(\mathsf{sid}, \mathsf{withraw}, x)$ from party U_i and C, party C

1. Assert($x \in \mathbb{N}$, $\mathsf{ActBalance}_i \geq x$ and $i \in [n]$).
2. $\mathsf{ActBalance}_i \mathrel{-}= x$.
3. Set $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{withraw}, i, x)$

History. Upon receiving $(\mathsf{sid}, \mathsf{history})$ from party P_i and C:

Send $(\mathsf{log}, \mathsf{log}_i)$ to P_i.

Audit. [**Iftach's Note: Later**]

3.2 The Protocol

Protocol 3.2 ($\Pi_{\mathsf{ConfTrans}}$: Confidential transactions)**.**

Parties: Issuer I, chain-holder C and users $\mathsf{U}_1, \ldots, \mathsf{U}_n$.

Paramters: 1^{κ_c}.

Subprotocols: See below.

Protocol 3.3 ($\Pi_{\mathsf{ConfTrans}}$.Init)**.**

Participating parties. All parties.

Operation:

1. P_i, for all $i \in [n]$,
   (a) Set $(pk_i, sk_i) \stackrel{R}{\leftarrow} \mathsf{KeyGen}(1^{\kappa_c})$.
   (b) Store sk_i.
   (c) Send pk_i to all parties.
2. All parties store $\{pk_i\}_{i \in [n]}$.
3. C:

(a) For all $i \in [n]$: Set $B_i \xleftarrow{R} \mathrm{Enc}_{pk_i}(0)$ and $H_i \leftarrow \emptyset$.
(b) Set $\mathsf{log} \leftarrow \emptyset$.

Protocol 3.4 ($\Pi_{\mathsf{ConfTrans}}$.Issue)**.**

Participating parties. I and C.

C's input. sid, $x \in \mathbb{N}$ and $i \in [n]$.

Operation:

1. I: Send (x, i) to C.
2. C: Set $H_i \cup= \{\mathsf{sid}, \mathsf{issue}, (\mathrm{Enc}_{pk_i}(x))\}$.
3. C: Set $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{issue}, x, i)$.

Protocol 3.5 ($\Pi_{\mathsf{ConfTrans}}$.Transfer)**.**

Participating parties. P_s and C.

Proof's systems: $\Pi^{\mathsf{pos}}.\Pi^{\mathsf{lrg}}$

Common input. $d \in [n]$.

P_s's private input. $x \in \mathbb{N}$.

Operation:

1. P_s:
   (a) $X \xleftarrow{R} \mathrm{Enc}_{pk_d}(x; r)$ for $r \xleftarrow{R} \{0,1\}^{\kappa_{\mathsf{c}}}$.
   (b) $\pi^{\mathsf{pos}} \xleftarrow{R} \mathsf{P}^{\mathsf{lrg}}((pk_d, X), (x, r))$.
   (c) $\pi^{\mathsf{lrg}} \xleftarrow{R} \mathsf{P}^{\mathsf{lrg}}((pk_s, pk_d, B_i, X), (sk_s, x, r))$.
   (d) Send $(X, \pi^{\mathsf{pos}}, \pi^{\mathsf{lrg}})$ to C.
2. C:
   (a) $\mathsf{V}^{\mathsf{pos}}(pk_d, X)$.
   (b) $\mathsf{V}^{\mathsf{lrg}}(pk_s, pk_d, B_i, X)$.
   (c) Set $H_d \cup= (\mathsf{sid}, s, X)$.
   (d) Set $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{transfer}, s, d)$.

Protocol 3.6 ($\Pi_{\mathsf{ConfTrans}}$.Update)**.**

Participating parties. P_i and C.

Operation: C

1. $B_i \mathrel{+}= \sum_{(\cdot,X)\in H_i} X$.
2. $H_i \leftarrow \emptyset$.
3. $\mathsf{log} \mathrel{+}= (\mathsf{sid}, \mathsf{rollover}, i)$

Protocol 3.7 ($\Pi_{\mathsf{ConfTrans}}$.History)**.**

Participating parties. P_i and C.

Operation: C sends (log, H_i) to P_i.