

Confidential Transactions

Theory Justification

Iftach Haitner*

August 4, 2025

Abstract

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

Contents

1	Introduction	2
2	Preliminaries	2
2.1	Notation	2
2.2	Homomorphic Encryption	2
3	The Confidential Transaction Protocols	2
3.1	The Ideal Functionality	2
3.2	The Protocol	3

*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 $[n] := \{1, \dots, n\}$ and $(n) := \{0, \dots, n\}$. For a relation \mathcal{R} , let $\mathcal{L}(\mathcal{R})$ denote its underlying language, i.e., $\mathcal{L}(\mathcal{R}) := \{x : \exists w : (x, w) \in \mathcal{R}\}$.

2.2 Homomorphic Encryption

An homomorphic encryption is a triplet $(\text{KeyGen}, \text{Enc}, \text{Dec})$ 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 $\text{Enc}_{sk}(x_0) + \text{Enc}_{pk}(x_1) \in \text{Supp}(\text{Enc}_{sk}(x_0 + x_1 \bmod q))$, 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 ($\mathcal{F}_{\text{ConfTrans}}$: Confidential transactions).

Parties: Issuer I , chain holder C and users U_1, \dots, U_n .

Init. Upon receiving init from all parties:

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

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

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

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

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

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

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

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

Withdraw. Upon receiving $(\text{sid}, \text{withdraw}, x)$ from party U_i and C, party C

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

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

Send (\log, \log_i) to P_i .

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

3.2 The Protocol

Throughout, we fix a security parameter κ and omit it from the notation. We also fix an homomorphic encryption scheme $(\text{KeyGen}, \text{Enc}, \text{Dec})$ over \mathbb{Z}_q with randomness domain \mathcal{D} . We require that $\text{Dec}_{sk}(A)$ outputs $(a; r)$ such that $A = \text{Enc}(a; r)$.

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

Parameters: $p_{\text{num}}, p_{\text{size}} \in \mathbb{N}$.

Parties: Issuer I, chain-holder C and users U_1, \dots, U_n .

Subprotocols: See below.

We use the of key-generation relation

$$\mathcal{R}_{\text{KeyGen}} = \{(pk, w) : \text{KeyGen}(w) = (\cdot, pk)\}.$$

Protocol 3.3 ($\Pi_{\text{ConfTrans-Init}}$).

Participating parties. All parties.

Protocols: $\Pi_{\mathcal{R}_{\text{KeyGen}}}^{\text{ZK-POK}}$.

Algorithms: KeyGen .

Operation:

1. P_i , for all $i \in [n]$:
 - (a) Set $(pk_i, sk_i) \xleftarrow{R} \text{KeyGen}(r_i)$ for $r_i \xleftarrow{R} \mathcal{D}$.
 - (b) Store sk_i .
 - (c) Let $\pi_i \xleftarrow{R} \text{PZK-POK}_{\mathcal{R}_{\text{KeyGen}}}(pk_i, r_i)$.
 - (d) Send (pk_i, π_i) to C .
2. C :
 - (a) Call $\{\text{V}_{\mathcal{R}_{\text{KeyGen}}}^{\text{ZK-POK}}(pk_i, \pi_i)\}_{i \in [n]}$. Abort and publish i , if the i^{th} proof is not verified.
 - (b) Store $\{pk_i\}_{i \in [n]}$.
3. C :
 - (a) Broadcast $\{p_i \leftarrow 0, P_i \xleftarrow{R} \text{Enc}_{pk_i}(0), B_i \leftarrow \emptyset\}_{i \in [n]}$.
 - (b) Broadcast $\log \leftarrow \emptyset$.

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

Participating parties. I and C .

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

Operation:

1. I : Send (x, i) to C .
2. C :
 - (a) Assert($x \in [p_{\text{size}}]$ and $p_i \leq p_{\text{num}}$)
 - (b) Set $P_i \mathrel{+}= \text{Enc}_{pk_i}(x)$.
 - (c) Publish $\log \cup = (\text{sid}, \text{issue}, x, i, P_i)$.

We use proofs for the following relations

$$\mathcal{R}_{\text{rp}} = \{((pk, A), (a, r)) : \text{Enc}_{pk}(a; r) = A \wedge a \in [p_{\text{size}}]\}$$

$$\mathcal{R}_{\text{eq}} = \{((pk_0, pk_1, A_0, A_1), (a, r_0, r_1)) : \forall i \in \{0, 1\} \text{Enc}_{pk_i}(a; r_i) = A_i\}$$

$$\mathcal{R}_{\text{lrg}} = \{((pk, A_0, A_1), (a, r_0, r_1)) : \forall i \in \{0, 1\} \text{Enc}_{pk}(a_i; r_i) = A_i\} \wedge a_1 - a_0 \in [q]$$

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

Participating parties. P_s and C .

Protocols: $\Pi_{\mathcal{R}_{\text{rp}}}^{\text{ZK-POK}}$.

Algorithms: Dec.

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

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

Operation:

1. P_s :

- (a) $X_d \xleftarrow{R} \text{Enc}_{pk_d}(x; r)$ for $r^d \xleftarrow{R} \mathcal{D}$.
- (b) $\pi^{rp} \xleftarrow{R} \text{P}_{\mathcal{R}_{rp}}^{\text{ZK-POK}}((pk_d, X_s, p_{\text{size}}), (x, r))$.
- (c) $X_s \xleftarrow{R} \text{Enc}_{pk_s}(x; r)$ for $r^s \xleftarrow{R} \mathcal{D}$.
- (d) $\pi^{\text{eq}} \xleftarrow{R} \text{P}_{\mathcal{R}_{\text{eq}}}^{\text{ZK-POK}}((pk_s, pk_d, X_s, X_s), (x, r_s, r_d))$.
- (e) $(b, r^b) \leftarrow \text{Dec}_{sk_s}(B_s)$.
- (f) $\pi^{\text{lrg}} \xleftarrow{R} \text{P}_{\mathcal{R}_{\text{lrg}}}^{\text{ZK-POK}}((pk_s, X_s, B_s), (x, r_s, r_b))$.
- (g) Send $(X_s, X_d, \pi^{rp}, \pi^{\text{eq}}, \pi^{\text{lrg}})$ to C .

2. C :

- (a) Call $\text{V}_{\mathcal{R}_{rp}}^{\text{ZK-POK}}((pk_d, X_s, p_{\text{size}}), \pi^{rp})$, $\text{V}_{\text{eq}}^{\text{ZK-POK}}((pk_s, pk_d, X_s, X_s), \pi^{\text{eq}})$ and $\text{V}_{\mathcal{R}_{\text{lrg}}}^{\text{ZK-POK}}((pk_s, X_s, B_s), \pi^{\text{lrg}})$.
- (b) Set $U_s \leftarrow X_s$.
- (c) Set $P_d \leftarrow X_d$.
- (d) Publish $\log \cup = (\text{sid}, \text{transfer}, s, d, U_s, P_d)$.

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

Participating parties. P_i and C .

Operation: C

- 1. $B_i \leftarrow \sum_{(\cdot, X) \in H_i} X$.
- 2. $H_i \leftarrow \emptyset$.
- 3. $\log \leftarrow (\text{sid}, \text{rollover}, i)$

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

Participating parties. P_i and C .

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