Confidential Transactions
Theory Justification

Iftach Haitner[*]

August 5, 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 $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ 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 $\mathsf{Enc}_{sk}(x_0) + \mathsf{Enc}_{pk}(x_1) \in \operatorname{Supp}(\mathsf{Enc}_{sk}(x_0 + x_1 \bmod q)$, 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 holder C and users $\mathsf{U}_1, \ldots, \mathsf{U}_n$.

Init. Upon receiving init from all parties:

1. For each $i \in [n]$: $\mathsf{AvlBlance}_i, \mathsf{PndBalance}_i \leftarrow 0$ 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($x \in \mathbb{N}$, $\mathsf{AvlBlance}_s \geq x$ and $s, d \in [n]$).
2. $\mathsf{AvlBlance}_s \mathrel{-}= x$.

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 $\mathsf{AvlBlance}_i \mathrel{+}= \mathsf{PndBalance}_i$.
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($x \in \mathbb{N}$, $\mathsf{AvlBlance}_i \geq x$ and $i \in [n]$).
2. $\mathsf{AvlBlance}_i \mathrel{-}= x$.
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

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

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

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

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

Subprotocols: See below.

We use the of key-generation relation

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

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

Participating parties. All parties.

Proofs: $\Pi^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{KeyGen}}}$.

Algorithms: KeyGen.

Operation:

1. P_i, for all $i \in [n]$:

   (a) Set $(pk_i, sk_i) \xleftarrow{\text{R}} \mathsf{KeyGen}(r_i)$ for $r_i \xleftarrow{\text{R}} \mathcal{D}$.

   (b) Store sk_i.

   (c) Let $\pi_i \xleftarrow{\text{R}} \mathsf{P}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{KeyGen}}}(pk_i, r_i)$.

   (d) Send (pk_i, π_i) to C.

2. C:

   (a) Call $\{\mathsf{V}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{KeyGen}}}(pk_i, \pi_i)\}_{i\in[n]}$. Abort and publish i, if the i^{th} proof is not verified.

   (b) Store $\{pk_i\}_{i\in[n]}$.

3. C:

   (a) Broadcast $\{p_i \leftarrow 0, P_i \xleftarrow{\text{R}} \mathsf{Enc}_{pk_i}(0), B_i \leftarrow \emptyset\}_{i\in[n]}$.

   (b) Broadcast $\mathsf{log} \leftarrow \emptyset$.

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:

   (a) Assert($x \in [p_{\mathsf{size}}]$ and $p_i \leq p_{\mathsf{num}}$)

   (b) Set $P_i \mathrel{+}= \mathsf{Enc}_{pk_i}(x)$).

   (c) Publish $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{issue}, x, i, P_i)$.

We use proofs for the following relations

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

I.e., encryption of values in $[p_{\mathsf{size}}]$.

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

I.e., encryptions of the same pair under different public keys.

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

I.e., encryptions of the pair of values (a_0, a_1), under the same public key, with $a_1 \geq a_0$.

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

Participating parties: P_s and C.

Proofs: $\Pi^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{rp}}}$.

Algorithms: Dec.

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

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

Operation:

1. P_s:
   (a) $X_d \xleftarrow{\mathrm{R}} \mathsf{Enc}_{pk_d}(x; r)$ for $r^d \xleftarrow{\mathrm{R}} \mathcal{D}$.
   (b) $\pi^{\mathsf{rp}} \xleftarrow{\mathrm{R}} \mathsf{P}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{rp}}}((pk_d, X_s, p_{\mathsf{size}}), (x, r))$.
   (c) $X_s \xleftarrow{\mathrm{R}} \mathsf{Enc}_{pk_s}(x; r)$ for $r^s \xleftarrow{\mathrm{R}} \mathcal{D}$.
   (d) $\pi^{\mathsf{eq}} \xleftarrow{\mathrm{R}} \mathsf{P}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{eq}}}((pk_s, pk_d, X_s, X_s), (x, r_s, r_d))$.
   (e) $(b, r^b) \leftarrow \mathsf{Dec}_{sk_s}(B_s)$.
   (f) $\pi^{\mathsf{lrger}} \xleftarrow{\mathrm{R}} \mathsf{P}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{lrger}}}((pk_s, X_s, B_s), (x, r_s, r_b))$.
   (g) Send $(X_s, X_d, \pi^{\mathsf{rp}}, \pi^{\mathsf{eq}}, \pi^{\mathsf{lrger}})$ to C.
2. C:
   (a) Call $\mathsf{V}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{rp}}}((pk_d, X_s, p_{\mathsf{size}}), \pi^{\mathsf{rp}})$,
   $\mathsf{V}^{\mathsf{ZK\text{-}POK}}_{\mathsf{eq}}((pk_s, pk_d, X_s, X_s), \pi^{\mathsf{rp}})$ and $\mathsf{V}^{\mathsf{ZK\text{-}POK}}_{\mathcal{R}_{\mathsf{lrger}}}((pk_s, X_s, B_s), \pi^{\mathsf{rp}})$.
   (b) Set $U_s \mathrel{-}= X_s$.
   (c) Set $P_d \mathrel{+}= X_d$.
   (d) Publish $\mathsf{log} \cup= (\mathsf{sid}, \mathsf{transfer}, s, d, U_s, P_d)$.

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

Participating parties. P_i and C.

Operation:
C:

1. $B_i \mathrel{+}= \mathrm{P}_i$.
2. $P_i \mathrel{-}= P_i$.
3. $\mathsf{log} \mathrel{+}= (\mathsf{sid}, \mathsf{rollover}, i, B_i, P_i)$

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

Participating parties. P_i and C.

Operation: C: send log to P_i.

4 The ElGammal-Based Additive Homomorphic Encryption

ee