MantaPay Trusted Setup Protocol Specification
v0.0.0

Francisco Hernández Iglesias, Todd Norton *

September 22, 2022

Abstract

We describe the protocol for the MantaPay trusted setup ceremony to generate prover and verifier keys for Groth16 ZK-SNARK proofs.

Contents

---

*ordered alphabetically.

1   Introduction

The MantaPay protocol (ref. to the specs) guarantees transaction privacy by using the Groth16 [1] Non-Interactive Zero-Knowledge Proving System (NIZK). In short, Groth16 is defined over a bilinear pairing of elliptic curves $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\tau$ over a prime field $\mathbb{F}_p$. Let $\phi \in \mathbb{F}_p^\ell$ denote the set of *public inputs*, $w \in \mathbb{F}_p^{m-\ell}$ the set of *witnesses*, whose knowledge we want to prove, and let $\tau \in (\mathbb{F}_p^*)^4$ be a set of randomly generated numbers known as the *simulation trapdoor*. Groth16 consists of four parts:

- $(\sigma, \tau) \leftarrow$ Setup: Randomly generates τ, from which it computes σ, which consists of elliptic curve points in $\mathbb{G}_1$ and $\mathbb{G}_2$. σ is to be understood as the proving and verifying keys for Groth16.

- $\pi \leftarrow$ Prove(σ, ϕ, w): Computes a proof of knowledge of w, π, for a given setup σ and public input ϕ.

- $0, 1 \leftarrow$ Verify(σ, ϕ, π): Checks whether the proof π is valid against the setup σ and the public input ϕ.

- $\pi \leftarrow$ Sim(τ, ϕ): Simulates a proof that will always be valid when verified against the setup σ corresponding to τ and the public input ϕ.

It is important to note that the Sim function is what makes the Groth16 protocol zero-knowledge: you can compute a valid proof π for any given setup σ and public input ϕ without knowledge of the witness w, provided that you have access to the simulation trapdoor τ. But Sim also makes Groth16 potentially insecure: if a malicious agent knew τ for a given σ, they could fabricate valid proofs for any statement regardless of its veracity.

The goal of the *trusted setup* is to provide a Groth16 setup σ in a secure way, i.e., in such a way that nobody has access to the trapdoor τ that was used to compute it.

2   Context

2.1   Circuit

Throughout this paper, by circuit we mean a *Rank-1 Constraint System (R1CS)*. It is defined as a system of equations over $\mathbb{F}_r$ of the form

$$\sum_{i=0}^{m} a_i u_{i,q} \cdot \sum_{i=0}^{m} a_i v_{i,q} = \sum_{i=0}^{m} a_i w_{i,q}, \qquad q = 1, \dots, n, \tag{1}$$

where $a_0 = 1$. This system of equations, in the context of zero-knowledge proofs, is to be understood as follows:

- The numbers $u_{i,q}, v_{i,q}, w_{i,q}$ are constants in $\mathbb{F}_r$ which represent the operations performed in an arithmetic circuit. Here constant means constant in the protocol, e.g. MantaPay will have a fixed set of $u_{i,q}, v_{i,q}, w_{i,q}$.

- The numbers $\phi = (a_1, \dots, a_\ell)$ are the public inputs. In MantaPay, these correspond to the TransferPost, excluding the proof.

- The numbers $w = (a_{\ell+1}, \dots, a_m)$ are the witnesses. In MantaPay, these correspond to the elements of the Transfer which are not part of the TransferPost.

- An R1CS defines the following binary relation

$$R = \left\{ (\phi, w) \mid \phi = (a_1, \dots, a_\ell), \ w = (a_{\ell+1}, \dots, a_m), \ (1) \text{ is satisfied} \right\} \subset \mathbb{F}_r^\ell \times \mathbb{F}_r^{m-\ell} \tag{2}$$

- The statements that can be proved in this terminology are of the form: for a given circuit (1) and public input ϕ, there exists[1] a witness w such that $(\phi, w) \in R$.

2.2   Quadratic Arithmetic Programmes

Quadratic Arithmetic Programmes (QAPs) give an alternative way to describe a circuit, equivalent to R1CS. A QAP is a system of polynomial equations of the form

$$\sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) \equiv \sum_{i=0}^{m} a_i w_i(X) \mod t(X), \tag{3}$$

where

---

[1]and I know

- $u_i(X), v_i(X), w_i(X) \in \mathbb{F}_r[X]$ are degree $n-1$ polynomials, and $t(X) \in \mathbb{F}_r[X]$ is a degree n polynomial, all of which are fixed for the protocol.

- The numbers $\phi = (a_1, \ldots, a_\ell)$ are the public inputs.

- The numbers $w = (a_{\ell+1}, \ldots, a_m)$ are the witnesses.

- A QAP defines the following binary relation

$$R = \left\{ (\phi, w) \mid \phi = (a_1, \ldots, a_\ell), \ w = (a_{\ell+1}, \ldots, a_m), \ (3) \text{ is satisfied} \right\} \subset \mathbb{F}_r^\ell \times \mathbb{F}_r^{m-\ell} \qquad (4)$$

- The statements that can be proved in this terminology are of the form: for a given circuit (3) and public input ϕ, there exists a witness w such that $(\phi, w) \in R$.

2.3 The Groth16 Setup function

Let us now recall how the Groth16 Setup function works in the MantaPay protocol. We start with

- The pairing curve BN254 [?], which consists of a triple of elliptic curves $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ and a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We fix generators g and h of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Note that $\mathrm{ord}(g) = \mathrm{ord}(h) = r$ is prime, and that $e(g, h)$ generates a subgroup of order r of $\mathbb{G}_T \cong F_{p^{12}}^*$.

- The MantaPay circuit, encoded as a QAP $\{u_i(X), v_i(X), w_i(X), t(X)\}$.

First, we randomly sample the simulation trapdoor $\tau = (\alpha, \beta, \delta, x) \leftarrow \mathbb{F}_r^*$. Second, we compute:

$$\sigma_1 = \left[\left(\alpha, \beta, \delta, \{x^i\}_{i=0}^{n-1}, \{\beta u_i(x) + \alpha v_i(x) + w_i(x)\}_{i=0}^\ell, \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\delta} \right\}_{i=\ell+1}^m, \left\{ \frac{x^i t(x)}{\delta} \right\}_{i=0}^{n-2} \right) \right]_1$$

$$\sigma_2 = \left[\left(\beta, \delta, \{x^i\}_{i=0}^{n-1} \right) \right]_2$$

where $[y]_1 = y \cdot g$ and $[y]_2 = y \cdot h$ for all $y \in \mathbb{F}_r$. Then we return $\sigma = (\sigma_1, \sigma_2)$. The goal of the trusted setup is to provide the provers and verifiers with σ without anyone knowing τ.

2.4 Phase 1

The parameters above split naturally into two classes. The first class consists of those which only depend on the elliptic curves and not on the circuit, i.e., on the QAP. These parameters, namely

$$KZG = ([\alpha]_1, [\beta]_1, \{[x^i]_1\}_{i=0}^{n-1}, [\beta]_2, \{[x^i]_2\}_{i=0}^{n-1}) \qquad (5)$$

are known as the *Kate–Zaverucha–Goldberg (KZG)* commitments. Since these parameters don't depend on the specifics of the circuit, we take them from the Perpetual Powers of Tau (PPoT) project [?], a multi-party computation similar to our trusted setup described below, with a 1-to-N security assumption. To ensure its soundness, we have personally verified each contribution to PPoT.

2.5 Phase 2

The rest of the Groth16 parameters do depend on the circuit, so we have the corresponding values for the MantaPay circuit. The rest of the paper is devoted to explaining in detail the multi-party computation known as the trusted setup ceremony which we perform to make sure the remaining Groth16 parameters are computed safely.

3 Requirements

3.1 Goals

- Anonymity: Pre-registration only requires a Twitter handle and an email address, none of which needs to be linked to your identity.

- Server coordination: The ceremony will be coordinated by a server, which will:
  - Manage the contribution queue for the participants, supporting priority tiers.

- Ensure that only pre-registered participants with valid signatures are allowed to take part in the ceremony.

  - Ensure that no more than one participant is contributing at a given time.

  - Ensure that no participant contributes more than once.

  - Inform the participants about their queue/contribution status.

  - Check the validity proof of the proposed contributions before adding them to the ceremony.

  - After a successful contribution, return its hash to the participant.

- Updatable: More contributions can be added to the ceremony at later stages if desired.

- Verifiable: The ceremony and all its contributions can be verified by independent auditors. We ensure that is the case by:

  - Publishing the transcript of the ceremony, including the validity proof and hash of each contribution.

  - Asking participants to publish their hashes in independent locations, e.g. Twitter.

  - Distributing an open-source verification library.

3.2   Non-Goals

- Permisionless: We require participants to pre-register to contribute to the ceremony. However, registration is open to anyone with a Twitter account and an email address.

- Support for independently computed contributions: The ceremony only supports those contributions done through the server with our client. However, the server code is open-source and the transcript of the ceremony is publicly available for audits.

4   Design

4.1   Ceremony Protocol

4.2   Server State Machine

4.3   Client State Machine

5   References

References

[1] Jens Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326. Springer, 2016.