Ciphertext-Commitment Equality Argument

1 Preliminaries

Basic notation. For two integers $n < m$, we write $[n, m]$ to denote the set $\{n, n+1, \ldots, m\}$. When $n = 1$, we simply write $[m]$ to denote the set $\{1, \ldots, m\}$. For any finite set S, we use $x \leftarrow_{\mathsf{R}} S$ to denote the process of sampling an element $x \in S$ uniformly at random. Unless specified otherwise, we use λ to denote the security parameter. We say that an algorithm is efficient if it runs in probabilistic polynomial time in the length of its input. We say that a function $f : \mathbb{N} \to \mathbb{N}$ is negligible if $f = o(1/n^c)$ for any positive integer $c \in \mathbb{N}$. Throughout the exposition, we use $\mathsf{poly}(\cdot)$ and $\mathsf{negl}(\cdot)$ to denote any polynomial and negligible functions respectively.

1.1 Discrete Log Relation Assumption

The discrete log relation assumption states that given a number of random group elements in $\mathbb{G}$, no efficient adversary can find a non-trivial relation on these elements.

Definition 1.1 (Discrete Log Relation)**.** Let $\mathbb{G} = \mathbb{G}(\lambda)$ be a group of prime order p. Then the *discrete log relation* assumption on $\mathbb{G}$ states that for any efficient adversary $\mathcal{A}$ and $n \geq 2$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr\left[\mathcal{A}(G_1, \ldots, G_n) \to a_1, \ldots, a_n \in \mathbb{Z}_p : \exists\ a_i \neq 0 \wedge \sum_{i \in [n]} a_i \cdot G = 0\right] = \mathsf{negl}(\lambda),$$

where $G_1, \ldots, G_n \leftarrow_{\mathsf{R}} \mathbb{G}$.

1.2 Rewinding Lemma

To prove security, we make use of the rewinding lemma. For the purpose of this document, we do not require the rewinding lemma in its full generality and therefore, we rely on the following simple variant from the work of Boneh et al. [**?**].

Lemma 1.2 (Rewinding Lemma)**.** *Let S, R, and T be finite, non-empty sets, and let X, Y, Y', Z, and Z' be mutually independent random variables such that*

- *X takes values in the set S,*
- *Y and Y' are each uniformly distributed over R,*
- *Z and Z' take values in the set T.*

Then for any function $f : S \times R \times T \to \{0, 1\}$, we have

$$\Pr\left[f(X, Y, Z) = 1 \wedge f(X, Y', Z') = 1 \wedge Y \neq Y'\right] \geq \varepsilon^2 - \varepsilon/N,$$

where $\varepsilon = \Pr[f(X, Y, Z) = 1]$ and $N = |R|$.

2 Zero-Knowledge Argument Definitions

In full generality, zero-knowledge argument systems can be defined with respect to any class of decidable languages. However, to simplify the presentation, we define argument systems with respect to CRS-dependent languages. Specifically, let $\mathcal{R} \subset \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ be an efficiently decidable ternary relation. Then a CRS-dependent language for a string $\rho \in \{0,1\}^*$ is defined as

$$\mathcal{L}_\rho = \{u \mid \exists\ w : (\rho, u, w) \in \mathcal{R}\}.$$

We generally refer to ρ as the common reference string, u as the instance of the langauge, and w as the witness for u.

For a class of CRS-dependent languages, an argument system consists of the following algorithms.

Definition 2.1 (Argument System)**.** A non-interactive argument system Π_{AS} for a CRS-dependent relation $\mathcal{R}$ consists of a tuple of efficient algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ with the following syntax:

- $\mathsf{Setup}(1^\lambda) \to \rho$: On input the security parameter λ, the setup algorithm returns a common reference string ρ.
- $\mathcal{P}(\sigma, u, w)$: The prover $\mathcal{P}$ is an interactive algorithm that takes in as input a common reference string σ, instance u, and witness w. It interacts with the verifier $\mathcal{V}$ according to the specification of the protocol.
- $\mathcal{V}(\sigma, u)$: The verifier $\mathcal{V}$ is an interactive algorithm that takes in as input a common reference string ρ and an instance x. It interacts with the prover $\mathcal{P}$ in the protocol and in the end, it either accepts (returns 1) or rejects (returns 0) the instance x.

We use $\langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u) \rangle = 1$ to denote the event that the verifier $\mathcal{V}$ accepts the instance of the protocol. We use $\langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u) \rangle \to \mathsf{tr}$ to denote the communication transacript between the prover $\mathcal{P}$ and verifier $\mathcal{V}$ during a specific execution of the protocol.

An argument system must satisfy a correctness and two security properties. The correctness property of an argument system is generally referred to as *completeness*. It states that if the prover $\mathcal{P}$ takes in as input a valid instance-witness tuple $(\rho, u, w) \in \mathcal{R}$ and follows the protocol specification, then it must be able to convince the verifier to accept.

Definition 2.2 (Completeness)**.** Let Π_{AS} be a proof system for a relation $\mathcal{R}$. Then we say that Π_{AS} satisfies perfect completeness if for any $(u, w) \in \mathcal{R}$, we have

$$\Pr\left[\langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u) \rangle = 1\right] = 1,$$

where $\rho \leftarrow \mathsf{Setup}(1^\lambda)$.

The first security property that an argument system must satisfy is *soundness*, which can be defined in a number of ways. In this work, we work with *computational witness-extended emulation* as presented in Bulletproofs [**?**].

Definition 2.3 (Soundness [**?**, **?**, **?**])**.** Let Π_{AS} be a proof system for a relation $\mathcal{R}$. Then we say that Π_{AS} satisfies *witness-extended emulation* soundness if for all deterministic polynomial time $\mathcal{P}^*$,

there exists an efficient emulator $\mathcal{E}$ such that for all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\left| \begin{array}{c} \Pr\left[\mathcal{A}_2(\mathsf{tr}) = 1 \;\middle|\; \begin{array}{c} \rho \leftarrow \mathsf{Setup}(1^\lambda), (u, \mathsf{st}) \leftarrow \mathcal{A}_1(\rho), \\ \mathsf{tr} \leftarrow \langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle \end{array}\right] - \\ \Pr\left[\mathcal{A}_2(\mathsf{tr}) = 1 \wedge (\mathsf{tr} \text{ accepting } \Rightarrow (\rho, u, w) \in \mathcal{R}) \;\middle|\; \begin{array}{c} \rho \leftarrow \mathsf{Setup}(1^\lambda), \\ (u, \mathsf{st}) \leftarrow \mathcal{A}_1(\rho), \\ (\mathsf{tr}, w) \leftarrow \mathcal{E}^{\mathcal{O}}(\rho, u) \end{array}\right] \end{array} \right| = \mathsf{negl}(\lambda),$$

where the oracle is defined as $\mathcal{O} = \langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$. The oracle $\mathcal{O}$ allows the emulator $\mathcal{E}$ to rewind the protocol to a specific point and resume the protocol after reprogramming the verifier with fresh randomness.

Traditionally, the soundness condition for an argument system of knowledge requires that there exists an extractor that can use its rewinding capability to extract a valid witness from any accepting transcript of the protocol that is produced by a dishonest prover $\mathcal{P}^*$. The witness-extended emulation strengthens this traditional definition by requiring that the extractor (emulator) not only successfully extracts a valid witness, but also produces (emulates) a valid transcript of the protocol for which the verifier accepts. The value st in the definition above can be viewed as the internal state of $\mathcal{P}^*$, which can also be its randomness.

The second security property that we require from an argument system is the zero-knowledge property. All argument systems that we rely on in the `ZK-Token` program are public coin protocols that we ultimately convert into a non-interactive protocol. Therefore, we rely on the standard zero-knowledge property against honest verifiers.

Definition 2.4 (Zero-Knowledge)**.** Let Π_{AS} be a proof system for a relation $\mathcal{R}$. Then we say that Π_{AS} satisfies *honest verifier* zero-knowledge if there exists an efficient simulator $\mathcal{S}$ such that for all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we have

$$\Pr\left[(\rho, u, w) \in \mathcal{R} \wedge \mathcal{A}_1(\mathsf{tr}) = 1 \;\middle|\; \begin{array}{c} \rho \leftarrow \mathsf{Setup}(1^\lambda), (u, w, \tau) \leftarrow \mathcal{A}_2(\rho), \\ \mathsf{tr} \leftarrow \langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u; \tau) \rangle \end{array}\right]$$

$$= \Pr\left[(\rho, u, w) \in \mathcal{R} \wedge \mathcal{A}_1(\mathsf{tr}) = 1 \;\middle|\; \begin{array}{c} \rho \leftarrow \mathsf{Setup}(1^\lambda), \\ (u, w, \tau) \leftarrow \mathcal{A}_2(\rho), \\ \mathsf{tr} \leftarrow \mathcal{S}(u, \tau) \end{array}\right],$$

where ρ is the public coin randomness used by the verifier.

3 Argument System Description

At the start of the protocol, the prover and verifier have access to a twisted ElGamal ciphertext and a Pedersen commitment. The prover's goal is to convince the verifier that it knows a secret key and a Pedersen opening such that the ciphertext and commitment decode to the same message. Formally, the language that is captured by the protocol is specified as follows:

$$\mathcal{L}^{\mathsf{eq\text{-}comm}}_{G,H} = \left\{ \begin{array}{c} u = (P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}}) \in \mathbb{G}^4, \\ w = (s, x, r) \in \mathbb{Z}_p^3 \end{array} \;\middle|\; \begin{array}{c} s \cdot P_{\mathsf{EG}} = H \wedge C_{\mathsf{EG}} - s \cdot D_{\mathsf{EG}} = x \cdot G \\ \wedge\; C_{\mathsf{Ped}} = x \cdot G + r \cdot H \end{array} \right\}.$$

The language $\mathcal{L}_{G,H}^{\mathsf{eq\text{-}comm}}$ is specified by two group elements $G, H \in \mathbb{G}$ that define the ElGamal encryption and Pedersen commitments. The group element P_{EG} corresponds to a public key in the twisted ElGamal encryption scheme and the field element s corresponds to its secret key. The elements $C_{\mathsf{EG}}, D_{\mathsf{EG}}$ correspond to a twisted ElGamal ciphertext and C_{Ped} corresponds to an additional Pedersen commitment. If $\mathsf{ct} = (C_{\mathsf{EG}}, D_{\mathsf{EG}})$ and C_{Ped} encode the same message x, then we must have $C_{\mathsf{EG}} - s \cdot D_{\mathsf{EG}} = x \cdot G$ and $C_{\mathsf{Ped}} = x \cdot G + r \cdot H$. The argument system for the language $\mathcal{L}_{G,H}^{\mathsf{eq\text{-}comm}}$ is specified as follows:

Prover(x, w)		**Verifier**(x)
$y_s \leftarrow_{\mathsf{R}} \mathbb{Z}_p$		
$y_x \leftarrow_{\mathsf{R}} \mathbb{Z}_p$		
$y_r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$		
$Y_0 \leftarrow y_s \cdot P_{\mathsf{EG}}$		
$Y_1 \leftarrow y_x \cdot G + y_s \cdot D_{\mathsf{EG}}$		
$Y_2 \leftarrow y_x \cdot G + y_r \cdot H$		
	$\xrightarrow{Y_0, Y_1, Y_2}$	
		$c \leftarrow_{\mathsf{R}} \mathbb{Z}_p$
	$\xleftarrow{c}$	
$z_s \leftarrow c \cdot s + y_s$		
$z_x \leftarrow c \cdot x + y_x$		
$z_r \leftarrow c \cdot r + y_r$		
	$\xrightarrow{z_s, z_x, z_r}$	
		$z_s \cdot P_{\mathsf{EG}} \stackrel{?}{=} c \cdot H + Y_0$
		$z_x \cdot G + z_s \cdot D_{\mathsf{EG}} \stackrel{?}{=} c \cdot C_{\mathsf{EG}} + Y_1$
		$z_x \cdot G + z_r \cdot H \stackrel{?}{=} c \cdot C_{\mathsf{Ped}} + Y_2$

As in the zero-balance argument protocol, the equality protocol follows a standard sigma protocol structure where the prover first samples random field eleemnts y_s, y_x, y_r. It then commits to these elements by sending $Y_0 = y_s \cdot P_{\mathsf{EG}}$, $Y_1 = y_x \cdot G + y_s \cdot D_{\mathsf{EG}}$, and $Y_2 = y_x \cdot G + y_r \cdot H$. Upon receiving a random challenge c, it provides the verifier with the masked secret key $z_s = c \cdot s + y$, $z_x = c \cdot x + y_x$, and $z_r = c \cdot r + y_r$. Finally, the verifier tests the three relations associated with $\mathcal{L}_{G,H}^{\mathsf{eq\text{-}comm}}$ using the masked secret key z, and the committed values Y_0, Y_1, and Y_2.

The equality argument system above satisfies all the correctness and security properties that are specified in Section 2. We formally state these properties in the following theorems.

Theorem 3.1 (Completeness)**.** *The ciphertext-commitment equality argument satisfies completeness 2.2.*

Theorem 3.2 (Soundness)**.** *Suppose that $\mathbb{G}$ is a prime order group for which the discrete log relation assumption (Definition 1.1) holds. Then the ciphertext-commitment equality argument satisfies witness-extended emulation soundness 2.3.*

Theorem 3.3 (Zero-Knowledge)**.** *The ciphertext-commitment equality argument satisfies perfect honest verifier zero-knowledge 2.4.*

4 Proof of Security

4.1 Proof of Theorem 3.1

to prove completeness, let us fix any valid instance and witness for $\mathcal{L}_{G,H}^{\mathsf{eq\text{-}comm}}$: $P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}} \in \mathbb{G}$ and $s, x, r \in \mathbb{Z}_p$ such that

- $s \cdot P_{\mathsf{EG}} = H$
- $C_{\mathsf{EG}} - s \cdot D_{\mathsf{EG}} = x \cdot G$
- $C_{\mathsf{Ped}} = x \cdot G + r \cdot H$

Let y_s, y_x, y_r and c be any elements in $\mathbb{Z}_p$, and let

- $Y_0 = y_s \cdot P$, $Y_1 = y_x \cdot G + y_s \cdot D_{\mathsf{EG}}$, $Y_2 = y_x \cdot G + y_r \cdot H$
- $z_s = c \cdot s + y_s$, $z_x = c \cdot x + y_x$, $z_r = c \cdot r + y_r$

in an execution of the protocol. Then we have

$$\begin{aligned} z_s \cdot P_{\mathsf{EG}} &= (c \cdot s + y_s) \cdot P_{\mathsf{EG}} \\ &= c \cdot (s \cdot P_{\mathsf{EG}}) + y_s \cdot P_{\mathsf{EG}} \\ &= c \cdot H + Y_0 \end{aligned}$$

$$\begin{aligned} z_x \cdot G + z_s \cdot D_{\mathsf{EG}} &= (c \cdot x + y_x) \cdot G + (c \cdot s + y_s) \cdot D_{\mathsf{EG}} \\ &= c \cdot (x \cdot G + s \cdot D_{\mathsf{EG}}) + (y_x \cdot G + y_s \cdot D_{\mathsf{EG}}) \\ &= c \cdot C_{\mathsf{EG}} + Y_1 \end{aligned}$$

$$\begin{aligned} z_x \cdot G + z_r \cdot D_{\mathsf{EG}} &= (c \cdot x + y_x) \cdot G + (c \cdot r + y_r) \cdot H \\ &= c \cdot (x \cdot G + r \cdot H) + (y_x \cdot G + y_r \cdot H) \\ &= c \cdot C_{\mathsf{Ped}} + Y_2 \end{aligned}$$

As all the algebraic relations that the verifier checks hold, the proof is always accepted. Completeness follows.

4.2 Proof of Theorem 3.2

To prove soundness, we construct an emulator $\mathcal{E}$ that has oracle access to any malicious prover $\mathcal{P}^*$ and extracts a valid witness by rewinding $\mathcal{P}^*$ and simulating two execution of the zero-balance protocol with an honest verifier $\mathcal{V}$.

Let $(P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}})$ be an instance of the language $\mathcal{L}_{G,H}^{\mathsf{eq\text{-}comm}}$. We construct an emulator $\mathcal{E}$ that uses $\mathcal{P}^*$ to extract a valid witness as follows:

- The emulator $\mathcal{E}$ first executes $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$ to produce a transcript $\mathsf{tr} = (Y_0, Y_1, Y_2, c, z_s, z_x, z_r)$.
- Then, it rewinds the protocol to the point where the verifier $\mathcal{V}$ samples a random $c \leftarrow_{\mathsf{R}} \mathbb{Z}_p$. It programs $\mathcal{V}$ with fresh randomness such that $\mathcal{V}$ generates a new $c' \leftarrow \mathbb{Z}_p$ independently of the previous execution of the protocol.
- The emulator completes the second execution of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$, producing a new transcript $\mathsf{tr} = (Y_0, Y_1, Y_2, c, z'_s, z'_x, z'_r)$.
- If $c - c' = 0$, then the emulator aborts and returns $\perp$. Otherwise, it computes
  - $s \leftarrow (z_s - z'_s)/(c - c')$
  - $x \leftarrow (z_x - z'_x)/(c - c')$
  - $r \leftarrow (z_r - z'_r)/(c - c')$

  and returns (s, x, r) as the witness.

To complete the proof, we first bound the probability that $\mathcal{E}$ does not abort at the end of the two executions of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$. Then, we show that if $\mathcal{E}$ does not abort, then the extracted witness (s, x, r) is valid.

Abort probability. The emulator $\mathcal{E}$ aborts only when $c = c'$, which is dependent on the probability that $\mathcal{P}^*$ successfully convinces $\mathcal{V}$ at the end of the protocol. Let $\varepsilon_{\mathcal{P}^*}$ be the probability that $\mathcal{P}^*$ successfully convinces $\mathcal{V}$ in $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$. We bound the probability that $c = c'$ with $\varepsilon_{\mathcal{P}^*}$ using the rewinding lemma 1.2. Specifically, let us define the following random variables:

- Let X be the elements (Y_0, Y_1, Y_2) in the transcript of an execution of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$.
- Let Y and Y' be the values c and c' respectively in the two executions of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$.
- Let Z and Z' be the values (z_s, z_x, z_r) and (z'_s, z'_x, z'_r) respectively in the two executions of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$.
- Let $f(\mathsf{tr}) \to \{0, 1\}$ be the protocol verification function that returns 1 if tr is an accepting transcript and 0 otherwise.

Then, the rewiding lemma states that

$$\Pr\left[f(X, Y, Z) = 1 \wedge f(X, Y', Z') = 1 \wedge Y \neq Y'\right] \geq \varepsilon^2_{\mathcal{P}^*} - \varepsilon_{\mathcal{P}^*}/p.$$

By assumption, we have $1/p = \mathsf{negl}(\lambda)$. Therefore, if $\varepsilon_{\mathcal{P}^*}$ is non-negligible, then the probability that $\mathcal{E}$ aborts at the end of the two executions of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$ is non-negligible.

Witness validity. Now assume that the two executions of $\langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u) \rangle$ returns two accepting transcripts $\mathsf{tr} = (Y_0, Y_1, Y_2, c, z_s, z_x, z_r)$, $\mathsf{tr}' = (Y_0, Y_1, Y_2, c', z'_s, z'_x, z'_r)$, and that $\mathcal{E}$ does not abort and returns

- $s \leftarrow (z_s - z'_s)/(c - c')$
- $x \leftarrow (z_x - z'_x)/(c - c')$
- $r \leftarrow (z_r - z'_r)/(c - c')$

Since tr and tr' are accepting transcripts, we have

$$z_x \cdot P_{\mathsf{EG}} = c \cdot H + Y_0,$$
$$z'_x \cdot P_{\mathsf{EG}} = c' \cdot H + Y_0,$$

This means that $(z_x - z'_x) \cdot P_{\mathsf{EG}} = (c - c') \cdot H$ and hence, $s \cdot P_{\mathsf{EG}} = H$. Similarly, we have

$$z_x \cdot G + z_s \cdot D_{\mathsf{EG}} = c \cdot C_{\mathsf{EG}} + Y_1,$$
$$z'_x \cdot G + z'_s \cdot D_{\mathsf{EG}} = c' \cdot C_{\mathsf{EG}} + Y_1,$$

This means that $(z_x - z'_x) \cdot G + (z_s - z'_s) \cdot D_{\mathsf{EG}} = (c - c') \cdot C_{\mathsf{EG}}$ and hence, $x \cdot G + s \cdot D_{\mathsf{EG}} = C_{\mathsf{EG}}$. Finally, we have

$$z_x \cdot G + z_r \cdot H = c \cdot C_{\mathsf{Ped}} + Y_2,$$
$$z'_x \cdot G + z'_r \cdot H = c' \cdot C_{\mathsf{Ped}} + Y_2,$$

which means that $(z_x - z'_x) \cdot G + (z_r - z'_r) \cdot H = (c - c') \cdot C_{\mathsf{Ped}}$ and hence, $x \cdot G + r \cdot H = C_{\mathsf{Ped}}$.

We have shown that if $\mathcal{P}^*$ successfully convinces the verifier $\mathcal{V}$ for an instance $x = (P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}})$ with non-negligible probability, then the emulator $\mathcal{E}$ successfully extracts a valid witness (s, x, r). This completes the proof of soundness.

4.3 Proof of Theorem 3.3

Fix any elements $P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}} \in \mathbb{G}$ and $s, x, r \in \mathbb{Z}_p$ such that $C_{\mathsf{EG}} - s \cdot D_{\mathsf{EG}} = x \cdot G$ and $C_{\mathsf{Ped}} = x \cdot G + r \cdot H$. Let $\mathsf{tr}^* = (Y_0^*, Y_1^*, Y_2^*, c^*, z_s^*, z_x^*, z_r^*)$ be any accepting transcript. By the specification of the protocol, the probability that an honest execution of the protocol by the prover and the verifier results in the transcript tr^* is as follows:

$$\Pr\left[\left\langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u) \right\rangle \to \mathsf{tr} \wedge \mathsf{tr} = \mathsf{tr}^*\right] = 1/p^4.$$

To prove zero-knowledge, we define a simulator $\mathcal{S}$ that produces such distribution without knowledge of a valid witness s, x, and r.

$\mathcal{S}(P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}})$:

1. Sample $c, z_s, z_x, z_r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$
2. Set $Y_0 = z_x \cdot P_{\mathsf{EG}} - c \cdot H$
3. Set $Y_1 = z_x \cdot G + z_s \cdot D_{\mathsf{EG}} - c \cdot C_{\mathsf{EG}}$
4. Set $Y_2 = z_x \cdot G + z_r \cdot H - c \cdot C_{\mathsf{Ped}}$
5. Return $\mathsf{tr} = (Y_0, Y_1, Y_2, c, z_s, z_x, z_r)$

The simulator $\mathcal{S}$ returns a transcript that is uniformly random given that

- $z_x \cdot P_{\mathsf{EG}} = c \cdot H + Y_0$,
- $z_x \cdot G + z_s \cdot D_{\mathsf{EG}} = c \cdot C_{\mathsf{EG}} + Y_1$,
- $z_x \cdot G + z_r \cdot H = c \cdot C_{\mathsf{Ped}} + Y_2$.

As the variables Y_0, Y_1 and Y_2 are completely determined by c, z_s, z_x, z_r, we have

$$\Pr\left[\mathcal{S}(P_{\mathsf{EG}}, C_{\mathsf{EG}}, D_{\mathsf{EG}}, C_{\mathsf{Ped}}) \to \mathsf{tr} \wedge \mathsf{tr} = \mathsf{tr}^*\right] = 1/p^4,$$

for any fixed transcript tr^*. Zero-knowledge follows.