Ciphertext Validity 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. [1].

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 (Setup, Prove, Verify) with the following syntax:

- 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 [2].

Definition 2.3 (Soundness [3, 4, 2])**.** 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

$$
\left|
\begin{array}{l}
\Pr\left[\mathcal{A}_2(\mathsf{tr}) = 1 \;\middle|\; \begin{array}{l} \rho \leftarrow \mathsf{Setup}(1^\lambda), (u, \mathsf{st}) \leftarrow \mathcal{A}_1(\rho), \\ \mathsf{tr} \leftarrow \left\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \right\rangle \end{array}\right] - \\[2em]
\Pr\left[\mathcal{A}_2(\mathsf{tr}) = 1 \wedge (\mathsf{tr}\ \text{accepting}\ \Rightarrow (\rho, u, w) \in \mathcal{R}) \;\middle|\; \begin{array}{l} \rho \leftarrow \mathsf{Setup}(1^\lambda), \\ (u, \mathsf{st}) \leftarrow \mathcal{A}_1(\rho), \\ (\mathsf{tr}, w) \leftarrow \mathcal{E}^{\mathcal{O}}(\rho, u) \end{array}\right]
\end{array}
\right| = \mathsf{negl}(\lambda),
$$

where the oracle is defined as $\mathcal{O} = \left\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \right\rangle$. 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

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

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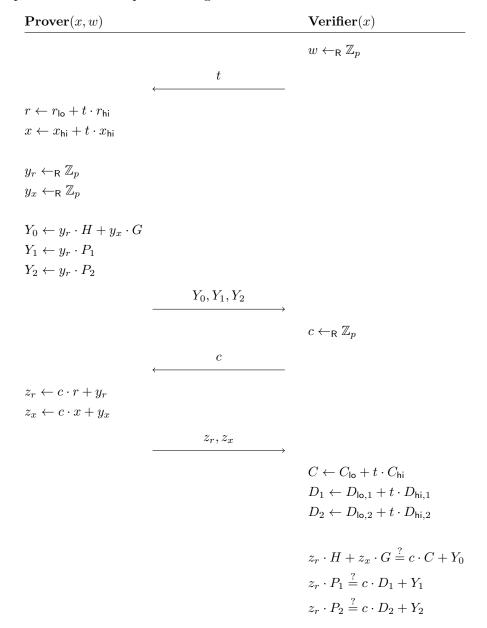$$

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

3 Ciphertext Validity Argument

At the start of a ciphertext valdity argument protocol, the prover and verifier have access to two joint ciphertexts $\mathsf{ct}_{\mathsf{lo}} = (C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2})$ and $\mathsf{ct}_{\mathsf{hi}} = (C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2})$. The prover's goal in the protocol is to convince the verifier that it knows valid randomness and message pairs $(r_{\mathsf{lo}}, x_{\mathsf{lo}})$ and $(r_{\mathsf{hi}}, x_{\mathsf{hi}})$ that each guarantee the validity of $(C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2})$ and $\mathsf{ct}_{\mathsf{hi}} = (C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2})$. Formally, the ciphertext-validity protocol captures the following language:

$$\mathcal{L}_{G,H}^{\text{ct-validity}} = \left\{ \begin{array}{l} u = (P_1, P_2, C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2}, C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2}) \in G^8, \\ w = (r_{\mathsf{lo}}, x_{\mathsf{lo}}, r_{\mathsf{hi}}, x_{\mathsf{hi}}) \in \mathbb{Z}_p^4 \end{array} \middle| \begin{array}{l} C_{\mathsf{lo}} = r_{\mathsf{lo}} \cdot H + x_{\mathsf{lo}} \cdot G \\ C_{\mathsf{hi}} = r_{\mathsf{hi}} \cdot H + x_{\mathsf{hi}} \cdot G \\ D_{\mathsf{lo},1} = r_{\mathsf{lo}} \cdot P_1 \\ D_{\mathsf{lo},2} = r_{\mathsf{lo}} \cdot P_2 \\ D_{\mathsf{hi},1} = r_{\mathsf{hi}} \cdot P_1 \\ D_{\mathsf{hi},2} = r_{\mathsf{hi}} \cdot P_2 \end{array} \right\}.$$

The formal specification of the protocol is given as follows:

Prover(x, w)	**Verifier**(x)
	$w \leftarrow_{\mathsf{R}} \mathbb{Z}_p$
$\xleftarrow{\quad t \quad}$	
$r \leftarrow r_{\mathsf{lo}} + t \cdot r_{\mathsf{hi}}$	
$x \leftarrow x_{\mathsf{hi}} + t \cdot x_{\mathsf{hi}}$	
$y_r \leftarrow_{\mathsf{R}} \mathbb{Z}_p$	
$y_x \leftarrow_{\mathsf{R}} \mathbb{Z}_p$	
$Y_0 \leftarrow y_r \cdot H + y_x \cdot G$	
$Y_1 \leftarrow y_r \cdot P_1$	
$Y_2 \leftarrow y_r \cdot P_2$	
$\xrightarrow{\quad Y_0, Y_1, Y_2 \quad}$	
	$c \leftarrow_{\mathsf{R}} \mathbb{Z}_p$
$\xleftarrow{\quad c \quad}$	
$z_r \leftarrow c \cdot r + y_r$	
$z_x \leftarrow c \cdot x + y_x$	
$\xrightarrow{\quad z_r, z_x \quad}$	
	$C \leftarrow C_{\mathsf{lo}} + t \cdot C_{\mathsf{hi}}$
	$D_1 \leftarrow D_{\mathsf{lo},1} + t \cdot D_{\mathsf{hi},1}$
	$D_2 \leftarrow D_{\mathsf{lo},2} + t \cdot D_{\mathsf{hi},2}$
	$z_r \cdot H + z_x \cdot G \overset{?}{=} c \cdot C + Y_0$
	$z_r \cdot P_1 \overset{?}{=} c \cdot D_1 + Y_1$
	$z_r \cdot P_2 \overset{?}{=} c \cdot D_2 + Y_2$

At the start of the protocol, the verifier sends the prover a challenge value $t \leftarrow_{\mathsf{R}} \mathbb{Z}_p$. The prover uses t to combine its witnesses $r \leftarrow r_{\mathsf{lo}} + t \cdot r_{\mathsf{hi}}$ and $x \leftarrow x_{\mathsf{hi}} + t \cdot x_{\mathsf{hi}}$. At this point of the protocol,

the prover and the verifier proceeds in a standard sigma protocol where the prover samples random scalar elements y_r, y_x and commits to them by sending $Y_0 = y_r \cdot H + y_x \cdot G$, $Y_1 = y_r \cdot P_1$, and $Y_2 = y_r \cdot P_2$ to the verifier. Upon receiving another challenge c, it provides the verifier with the masked randomness and message $z_r = c \cdot r + y_r$ and $z_x = c \cdot x + y_x$. Finally, the verifier tests the relations $z_r \cdot H + z_x \cdot G = c \cdot C + Y_0$, $z_r \cdot P_1 = c \cdot D_1 + Y_1$, and $z_r \cdot P_2 = c \cdot D_2 + Y_2$.

The ciphertext validity argument 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 validity 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 valdity argument satisfies witness-extended emulation soundness 2.3.*

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

We provide the formal proofs for these theorems in Section **??**.

4 Proof of Security

4.0.1 Proof of Theorem 3.1

To prove completeness, let us fix any valid instance and witness for $\mathcal{L}_{G,H}^{\mathsf{ct-validity}}$: $P_1, P_2, C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2}, C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2} \in \mathbb{G}$ and $r_{\mathsf{lo}}, x_{\mathsf{lo}}, r_{\mathsf{hi}}, x_{\mathsf{hi}} \in \mathbb{Z}_p$ such that

- $C_{\mathsf{lo}} = r_{\mathsf{lo}} \cdot H + x_{\mathsf{lo}} \cdot G$,
- $C_{\mathsf{hi}} = r_{\mathsf{hi}} \cdot H + x_{\mathsf{hi}} \cdot G$,
- $D_{\mathsf{lo},1} = r_{\mathsf{lo}} \cdot P_1$,
- $D_{\mathsf{lo},2} = r_{\mathsf{lo}} \cdot P_2$,
- $D_{\mathsf{hi},1} = r_{\mathsf{hi}} \cdot P_1$.
- $D_{\mathsf{hi},2} = r_{\mathsf{hi}} \cdot P_2$.

Let t, y_r, y_x, z_r, z_x be any elements in $\mathbb{Z}_p$ and let

- $Y_0 = y_r \cdot H + y_x \cdot G$,
- $Y_1 = y_r \cdot P_1$,
- $Y_2 = y_r \cdot P_2$,
- $z_r = c \cdot r + y_r$,
- $z_x = c \cdot x + y_x$,

in an execution of the protocol. Then we have

$$
\begin{aligned}
z_r \cdot H + z_x \cdot G &= (c \cdot r + y_r) \cdot H + (c \cdot x + y_x) \cdot G \\
&= c \cdot (r \cdot H + x \cdot G) + (y_r \cdot H + y_x \cdot G) \\
&= c \cdot \big((r_{\mathsf{lo}} + t \cdot r_{\mathsf{hi}}) \cdot H + (x_{\mathsf{lo}} + t \cdot x_{\mathsf{hi}}) \cdot G\big) + Y_0 \\
&= c \cdot \big(C_{\mathsf{lo}} + t \cdot C_{\mathsf{hi}}\big) \\
&= c \cdot C + Y_0
\end{aligned}
$$

5

$$\begin{aligned}
z_r \cdot P_1 &= (c \cdot r + y_r) \cdot P_1 \\
&= c \cdot (r \cdot P_1) + y_r \cdot P_1 \\
&= c \cdot \big((r_{\mathsf{lo}} + t \cdot r_{\mathsf{hi}}) \cdot P_1\big) + y_r \cdot P_1 \\
&= c \cdot (D_{\mathsf{lo},1} + t \cdot D_{\mathsf{hi},1}) + Y_1 \\
&= c \cdot D_1 + Y_1
\end{aligned}$$

$$\begin{aligned}
z_r \cdot P_2 &= (c \cdot r + y_r) \cdot P_2 \\
&= c \cdot (r \cdot P_2) + y_r \cdot P_2 \\
&= c \cdot \big((r_{\mathsf{lo}} + t \cdot r_{\mathsf{hi}}) \cdot P_2\big) + y_r \cdot P_2 \\
&= c \cdot (D_{\mathsf{lo},2} + t \cdot D_{\mathsf{hi},2}) + Y_2 \\
&= c \cdot D_2 + Y_2
\end{aligned}$$

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

4.0.2 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 four executions of the zero-balance protocol with an honest verifier $\mathcal{V}$.

Let $(P, C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2}, C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2})$ be an instance of the language $\mathcal{L}_{G,H}^{\mathsf{ct-validity}}$. We construct an emulator $\mathcal{E}$ that uses $\mathcal{P}^*$ to extract a valid witness. The emulator $\mathcal{E}$ rewinds the protocol at different stages. To simplify the presentation, we define a sub-emulator $\mathcal{E}_{\mathsf{inner}}$ that $\mathcal{E}$ uses as a subroutine to extract a valid witness. The sub-emulator $\mathcal{E}_{\mathsf{inner}}$ works as follows:

- The emulator $\mathcal{E}_{\mathsf{inner}}$ first executes $\big\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \big\rangle$ to produce a transcript $\mathsf{tr} = (w, Y_0, Y_1, Y_2, c, z_r, z_x)$.

- 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 $\big\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \big\rangle$, producing a new transcript $\mathsf{tr} = (t, Y_0, Y_1, c', z_r', z_x')$.

- If $c - c' = 0$, then the emulator aborts and returns $\bot$. Otherwise, it computes

    $- \; r \leftarrow (z_r - z_r')/(c - c')$
    $- \; x \leftarrow (z_x - z_x')/(c - c')$

    and returns (r, x).

We first bound the probability that $\mathcal{E}_{\mathsf{inner}}$ does not abort at the end of the two executions of $\big\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \big\rangle$. Then, we show that if $\mathcal{E}_{\mathsf{inner}}$ does not abort, then its output (r, x) satisfies

- $C = r \cdot H + x \cdot G$,
- $D_1 = r \cdot P_1$,

- $D_2 = r \cdot P_2$,

where $C = C_{\text{lo}} + t \cdot C_{\text{hi}}$, $D_1 = D_{\text{lo},1} + t \cdot D_{\text{hi},1}$, and $D_2 = D_{\text{lo},2} + t \cdot D_{\text{hi},2}$ in an execution of the protocol.

Abort probability of the sub-emulator. The emulator $\mathcal{E}_{\text{inner}}$ 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 $\langle \mathcal{P}^*(\rho, u, \text{st}), \mathcal{V}(\rho, u) \rangle$. 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 (w, Y_0, Y_1, Y_2) in the transcript of an execution of $\langle \mathcal{P}^*(\rho, u, \text{st}), \mathcal{V}(\rho, u) \rangle$.

- Let Y and Y' be the values c and c' respectively in the two executions of $\langle \mathcal{P}^*(\rho, u, \text{st}), \mathcal{V}(\rho, u) \rangle$.

- Let Z and Z' be the values (z_r, z_x) and (z_r', z_x') respectively in the two executions of $\langle \mathcal{P}^*(\rho, u, \text{st}), \mathcal{V}(\rho, u) \rangle$.

- Let $f(\text{tr}) \to \{0, 1\}$ be the protocol verification function that returns 1 if tr is an accepting transcript and 0 otherwise.

Then, the rewiding lemma states that

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

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

Output validity of sub-emulator. Now assume that the two executions of $\langle \mathcal{P}(\rho, u, w), \mathcal{V}(\rho, u) \rangle$ returns two accepting transcripts $\text{tr} = (t, Y_0, Y_1, Y_2, c, z_r, z_x)$, $\text{tr}' = (t, Y_0, Y_1, Y_2, c', z_r', z_x')$, and that $\mathcal{E}_{\text{inner}}$ does not abort and returns

- $r \leftarrow (z_r - z_r')/(c - c')$
- $x \leftarrow (z_x - z_x')/(c - c')$

Since tr and tr' are accepting transcripts, we have

$$z_r \cdot H + z_x \cdot G = c \cdot C + Y_0,$$

$$z_r' \cdot H + z_x' \cdot G = c' \cdot C + Y_0,$$

This means that $(z_r - z_r') \cdot H + (z_x - z_x') \cdot G = (c - c') \cdot C$ and hence, $r \cdot H + x \cdot G = C$. Similarly, we have

$$z_r \cdot P_1 = c \cdot D + Y_1,$$

$$z_r' \cdot P_1 = c' \cdot D + Y_1,$$

This means that $(z_r - z_r') \cdot P_1 = (c - c') \cdot D_1$, which means that $r \cdot P_1 = D_1$. The argument can be used to show that $r \cdot P_2 = D_2$.

Main emulator. For a language instance $u = (P, C_{\text{lo}}, D_{\text{lo},1}, D_{\text{lo},2}, C_{\text{hi}}, D_{\text{hi},1}, D_{\text{hi},2})$, the main emulator $\mathcal{E}$ executes two instances of the sub-emulator $\mathcal{E}_{\text{inner}}$ to obtain two outputs

7

- Let t be the verifier's first message in the protocol on the first execution of $\mathcal{E}_{\mathsf{inner}}$. The sub-emulator returns r and x such that

    - $C = r \cdot H + x \cdot G$,
    - $D_1 = r \cdot P_1$,
    - $D_2 = r \cdot P_2$,

    where $C = C_{\mathsf{lo}} + t \cdot C_{\mathsf{hi}}$, $D_1 = D_{\mathsf{lo},1} + t \cdot D_{\mathsf{hi},1}$, and $D_2 = D_{\mathsf{lo},2} + t \cdot D_{\mathsf{hi},2}$.

- Let t' be the verifier's first message in the protocol on the first execution of $\mathcal{E}_{\mathsf{inner}}$. The sub-emulator returns r' and x' such that

    - $C = r' \cdot H + x' \cdot G$,
    - $D_1 = r' \cdot P_1$,
    - $D_2 = r' \cdot P_2$,

    where $C' = C_{\mathsf{lo}} + t' \cdot C_{\mathsf{hi}}$, $D_1 = D_{\mathsf{lo},1} + t' \cdot D_{\mathsf{hi},1}$, and $D_2 = D_{\mathsf{lo},2} + t' \cdot D_{\mathsf{hi},2}$.

If $t = t'$ in the two executions, $\mathcal{E}$ aborts and returns $\bot$. Otherwise, the emulator returns the following:

- $r_{\mathsf{lo}} = (rt' - r't)/(t' - t)$ and $x_{\mathsf{lo}} = (xt' - x't)/(t' - t)$,

- $r_{\mathsf{hi}} = (r - r')/(t - t')$ and $x_{\mathsf{hi}} = (x - x')/(t - t')$.

To finish the proof, we 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 its output (r, x) is a valid witness.

Abort probability of the main emulator. The emulator $\mathcal{E}$ aborts only when $t = t'$, which is dependent on the probability that $\mathcal{E}_{\mathsf{inner}}$ successfully returns an output (r, x). Let $\varepsilon_{\mathcal{E}_{\mathsf{inner}}}$ be the probability that $\mathcal{E}_{\mathsf{inner}}$ successfully returns an output (r, x). We bound the probability that $t = t'$ with $\varepsilon_{\mathcal{E}_{\mathsf{inner}}}$ using the rewinding lemma. Specifically, let us define the following random variables:

- The variable $X = \varepsilon$ is an empty variable.

- Let Y and Y' be the values t and t' 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 in the two pairs of transcripts $\mathsf{tr} = (\mathsf{tr}_0, \mathsf{tr}_1)$ and $\mathsf{tr}' = (\mathsf{tr}'_0, \mathsf{tr}'_1)$ during $\mathcal{E}_{\mathsf{inner}}$'s executions of $\langle \mathcal{P}^*(\rho, u, \mathsf{st}), \mathcal{V}(\rho, u) \rangle$.

- Let $f(\mathsf{tr}) \rightarrow \{0, 1\}$ be the function that output 1 if $\mathcal{E}_{\mathsf{inner}}$ can successfully extract (r, x) from tr and 0 otherwise.

Then, the rewinding lemma states that

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

By assumption, we have $1/p = \mathsf{negl}(\lambda)$. Therefore, if $\varepsilon_{\mathcal{E}_{\mathsf{inner}}}$ is non-negligible, then the probability that $\mathcal{E}$ aborts at the end of the two executions of $\mathcal{E}_{\mathsf{inner}}$ is non-negligible.

Witness validity. Now assume that $\mathcal{E}$ does not abort after two executions of the protocol. Then it returns we have $t \neq t'$ and $\mathcal{E}$ returns

8

- $r_{\mathsf{lo}} = (rt' - r't)/(t' - t)$ and $x_{\mathsf{lo}} = (xt' - x't)/(t' - t)$,

- $r_{\mathsf{hi}} = (r - r')/(t - t')$ and $x_{\mathsf{hi}} = (x - x')/(t - t')$.

We show that $r_{\mathsf{lo}}, x_{\mathsf{lo}}, r_{\mathsf{hi}}, x_{\mathsf{hi}}$ make a valid witness for the ciphertext validity relation. By assumption on $\mathcal{E}_{\mathsf{inner}}$, the values r, x, r', x' satisfy the following relations:

$$r \cdot H + x \cdot G = C = C_{\mathsf{lo}} + t \cdot C_{\mathsf{hi}},$$

$$r' \cdot H + x' \cdot G = C = C_{\mathsf{lo}} + t' \cdot C_{\mathsf{hi}}.$$

Subtracting the two relations above, we have

$$(r - r') \cdot H + (x - x') \cdot G = (t - t') \cdot C_{\mathsf{hi}},$$

and hence, we have $(r - r')/(t - t') \cdot H + (x - x')/(t - t') \cdot G = C_{\mathsf{hi}}$.

Likewise, by assumption on $r_{\mathsf{lo}}, x_{\mathsf{lo}}, r_{\mathsf{hi}}, x_{\mathsf{hi}}$, we have

$$r \cdot P_1 = D_1 = D_{\mathsf{lo},1} + t \cdot D_{\mathsf{hi},1},$$

$$r' \cdot P_1 = D_1 = D_{\mathsf{lo},1} + t' \cdot D_{\mathsf{hi},1}.$$

Subtracting the two relations, we have

$$(r - r') \cdot P_1 = (t - t') \cdot D_{\mathsf{hi},1},$$

and hence, we have $(r - r')/(t - t') \cdot P_1 = D_{\mathsf{hi},1}$. Similar arguments shows that $r_{\mathsf{lo}} \cdot H + x_{\mathsf{lo}} \cdot G = C_{\mathsf{lo}}$, $r_{\mathsf{lo}} \cdot P_1 = D_{\mathsf{lo},1}$, $r_{\mathsf{lo}} \cdot P_2 = D_{\mathsf{lo},2}$, and $r_{\mathsf{hi}} \cdot P_2 = D_{\mathsf{hi},2}$. Soundness follows.

4.0.3 Proof of Theorem 3.3

Fix any elements $P, C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2}, C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2} \in \mathbb{G}$ and $r_{\mathsf{lo}}, x_{\mathsf{lo}}, r_{\mathsf{hi}}, x_{\mathsf{hi}} \in \mathbb{Z}_p$ such that the ciphertext validity relation hold. Let $\mathsf{tr}^* = (t^*, Y_0^*, Y_1^*, Y_2^*, c^*, z_r^*, z_x^*)$ 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 given by

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

To prove zero-knowledge, we define a simulator $\mathcal{S}$ that produces such distribution without knowledge of a valid witness $r_{\mathsf{lo}}, x_{\mathsf{lo}}, r_{\mathsf{hi}},$ and x_{hi}.

$\mathcal{S}(P, C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2}, C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2})$:

1. Sample $t, c, z_r, z_x, \leftarrow_{\mathsf{R}} \mathbb{Z}_p$
2. Let $C = C_{\mathsf{lo}} + t \cdot C_{\mathsf{hi}}$, $D_1 = D_{\mathsf{lo},1} + t \cdot D_{\mathsf{hi},1}$, and $D_2 = D_{\mathsf{lo},2} + t \cdot D_{\mathsf{hi},2}$
3. Set $Y_0 = z_r \cdot H + z_x \cdot G - c \cdot C$
4. Set $Y_1 = z_r \cdot P - c \cdot D_1$
5. Set $Y_2 = z_r \cdot P - c \cdot D_2$
6. Return $\mathsf{tr} = (w, Y_0, Y_1, c, z_r, z_x)$

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

- $z_r \cdot H + z_x \cdot G = c \cdot C + Y_0$,
- $z_r \cdot P_1 = c \cdot D_1 + Y_1$,
- $z_r \cdot P_2 = c \cdot D_2 + Y_1$,

where $C = C_{\mathsf{lo}} + w \cdot C_{\mathsf{hi}}$, $D_1 = D_{\mathsf{lo},1} + w \cdot D_{\mathsf{hi},1}$, and $D_2 = D_{\mathsf{lo},2} + w \cdot D_{\mathsf{hi},2}$. As the variables Y_0, Y_1, and Y_2 are completely determined by t, c, z_r, z_x, we have

$$\Pr\left[\mathcal{S}(P, C_{\mathsf{lo}}, D_{\mathsf{lo},1}, D_{\mathsf{lo},2}, C_{\mathsf{hi}}, D_{\mathsf{hi},1}, D_{\mathsf{hi},2}) \to \mathsf{tr} \wedge \mathsf{tr} = \mathsf{tr}^*\right] = 1/p^4,$$

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

References

[1] BONEH, D., DRIJVERS, M., AND NEVEN, G. Compact multi-signatures for smaller blockchains. In *International Conference on the Theory and Application of Cryptology and Information Security* (2018), Springer, pp. 435–464.

[2] BÜNZ, B., BOOTLE, J., BONEH, D., POELSTRA, A., WUILLE, P., AND MAXWELL, G. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)* (2018), IEEE, pp. 315–334.

[3] GROTH, J., AND ISHAI, Y. Sub-linear zero-knowledge argument for correctness of a shuffle. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2008), Springer, pp. 379–396.

[4] LINDELL, Y. Parallel coin-tossing and constant-round secure two-party computation. *Journal of Cryptology 16*, 3 (2003).