

Cryptographic Commitments on Anonymizable Data

Xavier Bultel*, Céline Chevalier †, Charlène Jojon*, Diandian Liu* and Benjamin Nguyen*

*LIFO, Université d'Orléans, INSA Centre Val de Loire, INRIA, France

†CRED, Université Panthéon-Assas, Paris II, France, DIENS, École normale supérieure, PSL University, CNRS, INRIA, Paris, France

Abstract—Local Differential Privacy (LDP) mechanisms consist of (locally) adding controlled noise to data in order to protect the privacy of their owner. In this paper, we introduce a new cryptographic primitive called LDP commitment. Usually, a commitment ensures that the committed value cannot be modified before it is revealed. In the case of an LDP commitment, however, the value is revealed after being perturbed by an LDP mechanism. Opening an LDP commitment therefore requires a proof that the mechanism has been correctly applied to the value, to ensure that the value is still usable for statistical purposes. We also present a security model for this primitive, in which we define the hiding and binding properties. Finally, we present a concrete scheme for an LDP staircase mechanism (generalizing the randomized response technique), based on classical cryptographic tools and standard assumptions. We provide an implementation in Rust that demonstrates its practical efficiency (the generation of a commitment requires just a few milliseconds).

On the application side, we show how our primitive can be used to ensure simultaneously privacy, usability and traceability of medical data when it is used for statistical studies in an open science context. We consider a scenario where a hospital provides sensitive patients data signed by doctors to a research center after it has been anonymized, so that the research center can verify both the provenance of the data (*i.e.* verify the doctors' signatures even though the data has been noised) and that the data has been correctly anonymized (*i.e.* is usable even though it has been anonymized).

1. Introduction

Individual's medical data presents the specific characteristic of being both used (i) by practitioners to help diagnose and propose treatments to patients (primary use) and (ii) by researchers as input data for clinical studies or statistical studies (secondary use). When data is used for treatment purposes, it must obviously be as precise as possible. However, in the context of large scale statistical studies, this data may benefit from additional privacy protection, such as (local) differential privacy (LDP) [1], [2], which is a standard model that provides formal statistical guarantees on the privacy of the data shared (described formally in Section 2). Indeed, there are currently many works where the use of such privacy preserving techniques are applied before performing data analysis (see [3] for a survey), quite often in the context of federated learning [4].

Our context is illustrated in Figure 1. In the following, the numbers in brackets refer to the different steps shown in this figure. Diana, a medical practitioner, is Alice and Bob's doctor. Bob is visiting her for a medical consultation, while Alice is monitored by various medical devices, configured by Diana. After a discussion, Diana and Alice on the one hand (1b), and Diana and Bob on the other (1a), agree on the value of the privacy parameters to apply, for the secondary use of their medical data [5]. Bob's data is produced by Diana through manual examination of her patient and can be signed. Alice's data is automatically extracted by the medical devices, and its authenticity can also be automatically signed. Both Alice and Bob's data, information on the anonymization process and parameters chosen are then sent to the datacenter of the hospital (managed by Helen) that Diana works for (2a-b). As expected, medical practitioners in the hospital use this data to give the correct treatments to Alice and Bob. At a later point in time (potentially when neither Diana, Bob nor Alice are available), Robin, a researcher from a different institution, interested in performing a large scale medical study, contacts Helen in order to obtain the hospital data. Helen transmits this authenticated data with local differential privacy constraints to Robin (3). Helen also provides Robin with a proof that she has correctly applied the differential privacy mechanism to the data signed by Diana, so that Robin and all those who will consult Robin's research results in an open science context are convinced that the data is usable and was produced legitimately. Depending on local privacy legislation (such as the European GDPR [6]), in the event of privacy violations, such as a data leak of Alice and Bob's exact medical information, an audit of Robin's research center may be performed by the relevant Data Protection Authority (Arthur). In this scenario, Robin must be able to prove his honesty and show that the data he owns was properly anonymized by presenting this proof (4). Robin can thus prove to Arthur that he is not responsible for the leak of Alice and Bob's exact data.

Therefore, we consider the following threat model, which is summarized in Fig 2. First, the doctor (Diana) and her patients (whom we will all consider as a single entity under the name Diana) trust Helen (the hospital manager) because they agree to provide her with their exact (non-anonymized) data in order to ensure optimal treatment for the patients. Robin (the researcher) and Helen trust Diana to sign the patient's real data, assuming that it is not in the patient's interest to give false information to the hospital. Data Protection Authority auditor (Arthur)

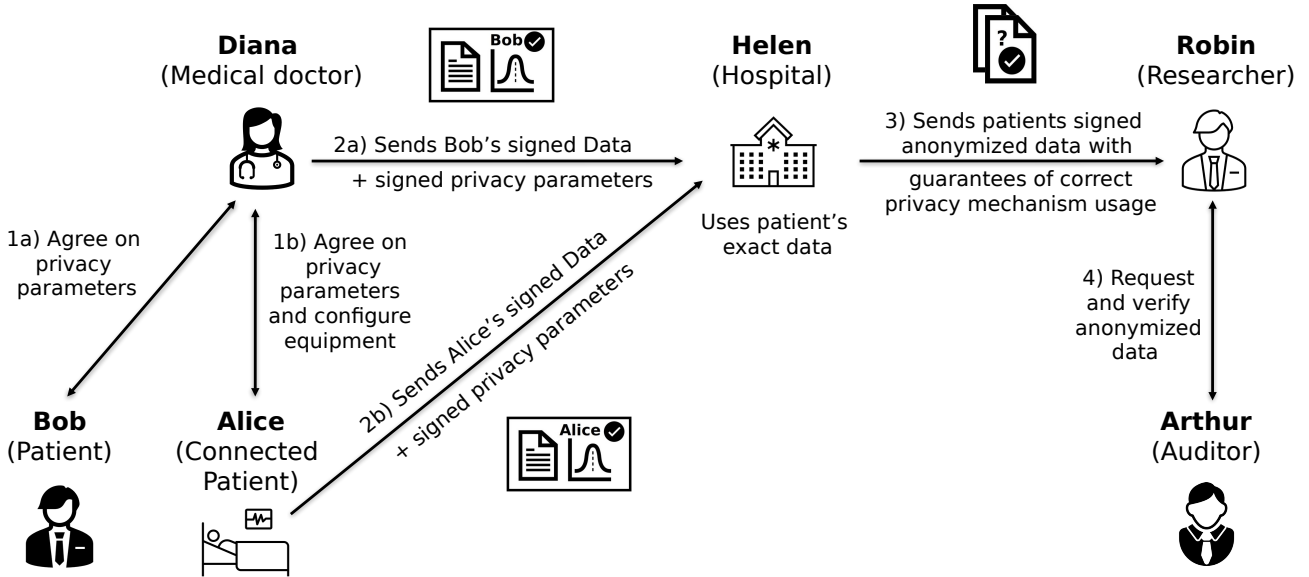


Figure 1. Medical research example

	Trusts Diana	Trusts Helen	Trusts Robin	Trusts Arthur
Diana	–	Yes	No	Yes
Helen	Yes	–	No	Yes
Robin	Yes	No	–	Yes
Arthur	Yes	No	No	–

Figure 2. Threat model.

also trusts Diana, since Arthur’s role is essentially to protect Diana’s privacy. On the other hand, Diana and Helen do not trust Robin, and do not want him to be able to view the unanonymized patient data. Robin does not trust Helen and wants guarantees that the data he receives from her has been correctly perturbed according to the chosen LDP mechanism. Finally, Arthur trusts neither Helen nor Robin, and must be able to verify that the data sent by the hospital and used by Robin in his research has been correctly anonymized, in accordance with the anonymization mechanism and the parameters chosen by Alice and Bob. As an audit authority, Arthur is obviously seen as a trusted entity.

Thus, in this article, we propose a new cryptographic setting in which sensitive medical data produced and signed by Diana, can both be used for medical treatment by Helen, and simultaneously be provided to Robin under differential privacy guarantees to perform statistical studies. The originality of the protocol is to support *re-producible* medical research by being able to *prove* that an anonymization protocol (with some given parameters) has correctly been executed on the real original data and to empower the patients through informed consent by letting them choose (after a discussion e.g. with Diana) what privacy parameters they wish to see applied to their data to be shared.

1.1. Our contributions

On a technical level, the first contribution of this paper is the formalization and security modelling of a new

primitive called *LDP Commitment*, which allows a user to commit a value in such a way that this value can be opened (potentially by another user) after the application of a local differential privacy mechanism. The verifier must be convinced that this mechanism has been applied correctly. Of course, only the anonymized data, and not the original data, is required to verify the correct opening of the commitment. Note that the LDP mechanism is inherently probabilistic and requires the use of a random seed. This seed cannot be fully chosen by the user committing the value and/or opening the commitment, as this would allow them to control the data after anonymization. Nor can the seed be fully chosen by the verifier, who could at best choose a seed that minimizes noise and at worst reverse the anonymization mechanism. It is therefore necessary that the choice of this seed should be shared between the data committer (at the time of commitment, it must not be possible for the opener to change it afterwards) and the verifier. Thus, if at least one of them chooses their nonce at random, the result is a perfectly random seed. In practice, in this setting the data owner has no interest whatsoever in not choosing a truly random nonce, since this guarantees the protection of their data.

First, we extend the classic hiding property for commitments (which guarantees that the committed data is kept private until the commitment is opened) to our primitive. We then define the LDP-hiding property, which ensures that the verifier obtains no more information about the committed value than the anonymized value. We give two formalisms for this property (real/simulation based and indistinguishability based): the first considers an adversary who must guess whether they received a real commitment or whether the commitment was simulated without using the value, and the second considers an adversary who chooses two values and must guess which of the two was committed and then opened. Unlike other indistinguishability games where the probability of success is compared to $1/2$ (the probability of guessing the hidden value among two), indistinguishability based LDP-hiding requires specifying the probability of success

of an adversary who would only exploit the anonymized message to find the original value. We give a simple formula to express this probability. We also show that classical hiding coupled with the real/simulation version implies the indistinguishability version.

We also extend the binding property (which ensures that a commitment can be open to unique value) to our primitive. We define the notion of LDP-binding, which catches an adversary trying not to apply the LDP mechanism correctly when they open the commitment. We model an adversary who commits a value and has to open it using a randomly generated seed. The scheme is considered to have the LDP-binding property if the distribution produced by this opening is the same as that produced by the application of the LDP mechanism on the value committed by the adversary.

Our second technical contribution is to propose a scheme for an LDP staircase mechanism (generalization of randomized response) and to prove its security in our model. Remarkably, our scheme uses only simple and efficient well-established standard cryptographic tools: it is based on the decisional Diffie–Hellman (DDH) assumption in a group of prime order and uses Schnorr-based zero-knowledge proofs. The complexity in terms of commitment size and computation time increases only logarithmically with the size of the set on which the data is chosen and the inverse of the probability of obtaining the real answer in the staircase. This makes the scheme very efficient in practice, as we show by analyzing the performance of a Rust implementation. In particular, the opening check is very efficient because it requires a constant number of exponentiations in the group, which is the most expensive operation of our algorithms by a huge margin. The commitment correction check takes a little longer but can be pre-computed before opening. In a nutshell, all our algorithms take a few milliseconds for standard parameters (when considering sets of a hundred responses) and less than 30 milliseconds in extreme cases (when considering sets of more than a billion responses) on a regular personal computer.

This new primitive allows us to address our problem: doctor Diana can sign the description of some medical value together with the commitment of this value (for example, “Self-assessment of pain, Commit(7)”, where Commit is an LDP commitment algorithm). Note that the commitment, which is signed, gives no information about the committed value (in our example 7). Moreover, this value is not needed to verify the signature. The value, the description, the signature, the commitment, and corresponding opening key are sent to Helen, the hospital manager, who stores them. If Helen wishes to transmit this value after applying the LDP mechanism, she sends the signature to a verifier with the opening of the commitment on the anonymized value. By verifying that the commitment has been opened correctly, verifier Robin is convinced that the LDP mechanism has indeed been applied to the anonymized value transmitted by Helen, and by verifying the signature on the commitment, Robin is convinced that the original committed value (before anonymization) does indeed come from Diana. However, Robin learns nothing about this original value. Thus, Helen cannot cheat on the application of the LDP mechanism to the value signed by Diana. Furthermore, in the event

of an audit by Arthur due to the leakage of a patient’s personal data, Arthur can consult the proof of opening of the commitment and the signature on the commitment to ensure that the data sent by Helen is correctly anonymized. Note that our implementation efficiency analysis takes into account the signature of the commitment.

1.2. Related work

In 2008, Ambainis, Jakobsson and Lipmaa first proposed the use of cryptographic tools to secure LDP mechanisms in [7]. Their work focuses on a randomized response mechanism where the response is a bit and the probability of giving that response depends on a parameter. They propose interactive protocols between a responder and a verifier where the verifier can check that the responder has correctly applied the LDP mechanism on their response, without learning any additional information about the exact response. Our work can be seen as an extension of this first work, where the response set is not limited to one bit and where the data can be committed before being revealed, potentially by an untrusted delegate. Note also that our protocol does not require any interaction between the parties.

In recent years, the growing interest in differential privacy and recent studies showing its critical weakness in the front of manipulation attacks [8], where respondents attempt to bias the results of statistics by manipulating their answers, have led to renewed interest in this approach. In [9], Kato *et al.* extends the work of Ambainis *et al.* [7] to other randomization mechanisms (still in an interactive setting where the data owner randomizes their data themselves). In [10], Biswas and Cormode extend the concept of verifiable differential privacy to the non-local case, where multiple entities commit their data, and a server computes a function on that data, reveals the anonymized result, and proves that it has correctly applied the differential privacy mechanism using an interactive zero-knowledge proof. They experiment their proposal by designing and implementing proofs for counting queries with the binomial DP mechanism [1]. Our work is aimed at a different application than in [10], since we are focusing on local differential privacy using a generalized randomized response mechanism. While it is therefore difficult to compare this work with ours, it can be noted that our protocol does not require interaction between the parties and is more efficient (their zero-knowledge proofs are interactive and take several minutes, whereas our commitment opening proofs take around ten milliseconds for the largest parameters). Moreover, we propose a more general and complete security model than in [10] that can be applied to solutions that are not unique zero-knowledge proof protocols.

Another line of work consists in using generic tools such as multiparty computation [11] or homomorphic encryption [12] to compute statistical functions from the private data of several users with differential privacy, either in a setting where some users are honest [13], or in a model where the curator is honest but curious [11], [14]. Our work differs from these works in that we focus on local differential privacy where data is committed by users and revealed (with differential privacy) later by another user, we propose specific efficient and non-interactive

protocols rather than using generic tools, and we only consider dishonest users.

In [15], the authors use generic zero-knowledge proofs for arithmetic circuits (zkSNARKs) to verify that a data item has been correctly anonymized using a local differential privacy mechanism. They use Circome [16] to implement the circuit corresponding to the randomized response mechanism (non-generalized, on a single bit) and the exponential mechanism on a set of 128 values. They use the Groth-16 [17] zkSNARKs (based on pairings in the generic group model) to experiment with their construction. The execution time for generating and verifying each proof takes around a second in a setting comparable to ours. Moreover, the authors mention that their proof could be used on data signed in an anonymous credential, resulting in a scheme similar to ours, however, this would require the use of a much more complex circuit than the one actually presented in their paper, making it possible to prove knowledge of a valid (hidden) signature on the (hidden) data used by the DP mechanism. Note that the efficiency of the Groth-16 proof [17] depends on the complexity of the circuit used: the number of exponentiations required for the proof grows linearly with the number of wires and multiplication gates in the proven arithmetic circuit. Adding the verification of a signature to the data used in the anonymization mechanism, and proving that this data is indeed the one that was committed, should therefore make the circuit much more complex, and therefore degrade the performance of this approach for our application. Our work offers a more efficient alternative (The execution time for generating and verifying each proof takes a few milliseconds), by designing a specific protocol rather than using generic tools. Moreover, [15] does not model and prove the security of their construction. We believe that our model could be used for this purpose.

Finally, several signature primitives allow delegating the power to modify/sign messages. Sanitizable signatures [18], [19] allow a delegate to modify certain parts of a signed message, but the restrictions on these modifications are either too specific [20] or too weak [18] to be used for differential privacy mechanisms. Functional signatures [21] and delegatable functional signatures [22] allow a user to sign messages that verify certain functions or predicates, policy-based signatures [23] allow a user to sign a message if it satisfies a certain policy, and homomorphic signatures [24] allow computations to be performed on signed data. These primitives could be considered for our problem, but their genericity implies a significant loss of efficiency (due to the use of heavy tools, such as proofs for circuits), whereas the aim of this work is to provide a simple and optimized solution, tailored for local differential privacy.

2. Background

In this section, we introduce our notations as well as the tools that we need, both from cryptography and (local) differential privacy.

We denote by $\llbracket n \rrbracket$ the set $\{1, \dots, n\}$ where n is an integer. By $x \leftarrow y$ we mean that the variable x takes a value y , by $x \leftarrow \text{Algo}(y)$ that the variable x takes a value outputted by algorithm Algo on input y , and by $r \stackrel{\$}{\leftarrow} S$ that

r is chosen from the uniform distribution on S . Finally $r_1, \dots, r_n \stackrel{\$}{\leftarrow} S$ means that $\forall i \in \llbracket n \rrbracket, r_i \stackrel{\$}{\leftarrow} S$. We use the acronym PPT for *probabilistic polynomial time*, s.t. for *such that* and λ denotes the security parameter.

Cryptographic tools. We recall the Decisional Diffie-Hellman (DDH) assumptions in a group $\mathbb{G} = \langle g \rangle$ of prime order p . The DDH assumption states that given a random $(x, y, z) \in (\mathbb{Z}_p^*)^3$, it is difficult for a PPT algorithm \mathcal{D} to distinguish between $(g^x, g^y, g^{x \cdot y})$ and (g^x, g^y, g^z) .

Definition 1 (Decisional Diffie-Hellman (DDH) assumption). *Let $\mathbb{G} = \langle g \rangle$ be a multiplicative group of prime order p . The decisional Diffie-Hellman (DDH) assumption states that there exists a negligible function ϵ_{DDH} such that for any PPT algorithm \mathcal{D} ,*

$$\begin{aligned} & |\Pr[(x, y) \stackrel{\$}{\leftarrow} (\mathbb{Z}_p^*)^2 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^{x \cdot y})] - \\ & \Pr[(x, y, z) \stackrel{\$}{\leftarrow} (\mathbb{Z}_p^*)^3 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^z)]| \leq \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

We recall the concept of Non-Interactive Zero Knowledge Proofs (NIZKP), that allows a prover to prove the knowledge of a witness matching a statement to a verifier.

Definition 2 (Non-Interactive Proofs (NIP) [25]). *Let \mathcal{R} be a binary relation and let \mathcal{L} be a language such that $s \in \mathcal{L} \Leftrightarrow (\exists w, (s, w) \in \mathcal{R})$. A Non-Interactive Proof (NIP) for \mathcal{L} is a couple of algorithms (NIP, NIPVerify) such that:*

NIP $\{w : (s, w) \in \mathcal{R}\}$. *This algorithm outputs a proof π .*
NIPVerify (s, π) . *This algorithm outputs a bit b .*

A Non-Interactive Zero Knowledge Proof (NIZKP) is a NIP having the following properties:

Correctness. *For any s, w, \mathcal{R} such that $(s, w) \in \mathcal{R}$ and $\pi \leftarrow \text{NIP}\{w : (s, w) \in \mathcal{R}\}$, $\text{NIPVerify}(s, \pi)$ returns 1.*

Extractability. *There exists a PPT knowledge extractor Ext and a negligible function ϵ_{ext} such that for any algorithm $\mathcal{A}^{\text{Sim}(\cdot)}(\lambda)$ having access to a simulator that forges signatures for chosen statement and that outputs a fresh pair (s, π) with $\text{NIPVerify}(s, \pi) = 1$, the extractor $\text{Ext}^{\mathcal{A}}(\lambda)$ outputs w such that $(s, w) \in \mathcal{R}$ having access to $\mathcal{A}(\lambda)$ with probability at least $1 - \epsilon(\lambda)$.*

Zero-knowledge. *A proof π leaks no information, i.e., there exists a polynomial time algorithm Sim (called the simulator) such that $\text{NIP}\{w : (s, w) \in \mathcal{R}\}$ and $\text{Sim}(s)$ follow the same probability distribution.*

Local Differential Privacy (LDP). Local differential privacy [2] (LDP) is a well accepted standard definition to measure the security (privacy) of randomized algorithms in the privacy domain. Intuitively, local differential privacy is a constraint on the distribution of the outcomes of the algorithm, when an attacker observes the output of a single execution of the algorithm. It is well adapted in the context of independent devices or individuals producing results that must present formal privacy guarantees. LDP can be seen as a non-centralized version of the differential privacy model [1]. Formally,

Definition 3 (Local Differential Privacy [2]). *A randomized algorithm \mathcal{A} satisfies ϵ -local differential privacy*

(LDP) where $\epsilon \in \mathbb{R}^{+*}$, if for any pair of input values $(x_1, x_2) \in \text{Domain}(\mathcal{A})$ and any possible output $y \in \text{Codomain}(\mathcal{A})$:

$$\Pr[\mathcal{A}(x_1) = y] \leq \exp(\epsilon) \cdot \Pr[\mathcal{A}(x_2) = y].$$

Note that the greater the value of ϵ (called *privacy budget* in differential privacy models), the lesser the privacy of the mechanism. Another important characteristic of LDP is the *utility* of the mechanism, which is often measured by the difference between two distributions (called *f-divergence*) [26]. Indeed, the authors of [26] have shown that there exists a class of optimal mechanisms (i.e. which maximize utility for a given ϵ value) called *staircase mechanisms*. These mechanisms satisfy the following constraints:

- the output domain size (noted $|\mathcal{Y}|$) is at most the input domain size (noted $|\mathcal{X}|$); and
- for all $y \in \mathcal{Y}$, and $x, x' \in \mathcal{X}$, $\left| \ln \frac{\Pr(\mathcal{A}(x=y))}{\Pr(\mathcal{A}(x'=y))} \right| \in \{0, \epsilon\}$ with $\epsilon > 0$.

This means that an optimal mechanism \mathcal{A} can only output values with two different probabilities p_1 and p_2 with $p_1 \geq p_2$ such that $p_1 = \exp(\epsilon) \cdot p_2$. Thus, for a mechanism with a bounded input domain and output domain as in our case, the simple optimal mechanism that we consider is the following (noted LDP in the rest of the article):

Definition 4 (Optimal mechanism LDP). For all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and for a given $\epsilon \in \mathbb{R}^{+*}$,

- LDP(x) outputs y with probability p_1 if $x = y$
- LDP(x) outputs y with probability p_2 if $x \neq y$
- $p_1 = \exp(\epsilon) \cdot p_2$

We show in Section 4.1 how to compute probabilities p_1 and p_2 given the size of the domain $|\mathcal{X}|$.

3. Formal Model for LDP Commitment

In this section we give a formal definition and security model for LDP commitments.

3.1. Formal Definition

As with standard cryptographic commitments, an LDP commitment allows a user to commit a secret value in such a way that this value can be opened (i.e. revealed) using a key. The user consulting the opened value can use an algorithm to check that the commitment has been correctly opened, i.e. that the opened value is indeed the one that was committed. In addition, an LDP commitment also depends on a public LDP mechanism. A user with the key can, from a random seed, use an alternative opening algorithm that reveals the anonymized value (to which the LDP mechanism was applied). In this case, the user also produces a proof that they have correctly applied the opening algorithm, without revealing anything more about the actual committed value. Finally, the commitment algorithm also produces a proof that the commitment was indeed generated by the LDP mechanism, ensuring that the released value is indeed the value that was committed to by applying the given LDP mechanism. This random seed must be chosen by someone other than the user opening the commitment (either by a trusted third party or

directly by the verifier), otherwise the user could choose the seed that reveals the value they chose instead of following the LDP mechanism distribution.

Definition 5 (LDP Commitment). A LDP Commitment (LDP-C) is a tuple of polynomial time algorithms (Setup, Commit, Open, OpenLDP, VerOpen, VerOpenLDP, VerCommit) such that:

- Setup(λ , LDP) : takes as input the security parameter λ , and a LDP mechanism LDP; returns a setup set.
- Commit(m, θ) : takes as input a message m and a random seed θ ; returns a secret key k , a commitment c and a proof π_* .
- VerCommit(c, π_*) : takes as input a commitment c and a proof π_* ; returns an acceptance bit b .
- Open(k, c) : takes as input a secret key k and a commitment c ; returns a message m and a proof π .
- VerOpen(c, m, π) : takes as input a commitment c , a message m , and a proof π ; returns an acceptance bit b .
- OpenLDP($k, c, \hat{\theta}$) : takes as input a secret key k , a commitment c , and a random seed $\hat{\theta}$; returns a message \hat{m} and a proof $\hat{\pi}$.
- VerOpenLDP($c, \hat{m}, \hat{\pi}, \hat{\theta}$) : takes as input a commitment c , a message \hat{m} , a proof $\hat{\pi}$, and a seed $\hat{\theta}$; returns an acceptance bit b .

3.2. Security Model

The natural properties of cryptographic commitments are *hiding*, which guarantees that the commitment does not reveal any information about the committed value before being opened, and *binding*, which guarantees that the commitment can only be opened in one way (by revealing the actual committed value). In our case, since there are several ways to open LDP commitments (with or without the LDP mechanism), we need to adapt these properties accordingly. Figure 3 shows the experiments corresponding to the different variations of these properties we define.

3.2.1. Hiding. First of all, we define Hiding in the classical way: an adversary who chooses two messages and receives a commitment for one of the two messages is unable to distinguish which message is being used (its probability of distinguishing is close to 1/2).

Definition 6 (Hiding). Let λ be a security parameter; LDP be a LDP mechanism, and P be a LDP-C. P is said to be hiding if for any PPT two-party algorithm \mathcal{A} , there exists a negligible function ϵ s.t.:

$$\left| \Pr \left[0 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{Hiding}}(\lambda) \right] - \Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{Hiding}}(\lambda) \right] \right| \leq \epsilon(\lambda),$$

where the Hiding experiment is given in Figure 3.

Note: As we use the standard notations from the security and differential privacy worlds, ϵ and $\epsilon(\lambda)$ represent different concepts. However, as we feel that their meaning is obvious from the context, we have kept these notations.

We then consider an adversary for whom we open a commitment with the algorithm OpenLDP. In this case,

the adversary is not expected to learn any more information about the original value than they learn from the anonymized value returned by the commitment opening algorithm OpenLDP. To model this, we first propose an experiment (ROS-LDP-Hiding) where the adversary has to guess whether they have received the biased value returned by OpenLDP (*real case* with $b = 0$), or whether they have received the biased value by the true LDP mechanism (*simulated case* with $b = 1$, in which case the proof of correction of the opening must be simulatable to the adversary). To do this, the adversary can choose the value committed, and can also choose the random seed used for the opening.

Definition 7 (Real Or Simulated (ROS) LDP-Hiding). *Let λ be a security parameter, LDP be a LDP mechanism, and P be a LDP-C. P is said to be ROS-LDP-hiding if for any three-party PPT algorithm \mathcal{A} , there exists a negligible function ϵ and a PPT simulator Sim that takes as input a message m and a commitment C , and that returns a simulated proof $\hat{\pi}$ s.t.:*

$$\left| \Pr \left[0 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{ROS-LDP-Hiding}}(\lambda) \right] - \Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{ROS-LDP-Hiding}}(\lambda) \right] \right| \leq \epsilon(\lambda),$$

(the ROS-LDP-Hiding experiment is given in Figure 3).

Another way of formalizing this property, which is closer to standard Hiding, is to commit a message drawn from two messages chosen by the adversary, open it with OpenLDP, and verify whether the adversary can distinguish which message has been committed. We define the corresponding security experiment as IND-LDP-Hiding. However, the opened message, although anonymized, can leak significant information about the actual committed message. This security property cannot therefore be defined as the fact that the difference between the probability that the adversary answers correctly and the probability that it is wrong is negligible, as is usually the case in indistinguishability experiments. For the same reasons, it is not possible to compare the probability of the adversary distinguishing the message to $1/2$. We therefore need to evaluate the probability that an adversary, acting optimally, would have of recovering the original message from a message anonymized by the given LDP mechanism only (without knowing the commitment). To do this, we define the algorithm OptiGuess, which takes as input two original messages (m_0, m_1) and an anonymized message \hat{m} computed by applying the LDP mechanism to one of these messages, then returns b when m_b is the most likely original message knowing only the anonymized message \hat{m} . From a technical point of view, the IND-LDP-Hiding experiment is parameterized by a bit b . When $b = 0$, we observe the probability of the adversary correctly distinguishing the original message, otherwise, when $b = 1$, we observe the optimal probability (given by OptiGuess) of finding the original message among the messages chosen by the adversary using only the anonymized value. The IND-LDP-hiding security is satisfied when the probability of success of \mathcal{A} when $b = 0$ is significantly higher than in the case $b = 1$.

Definition 8 (Indistinguishability (IND) LDP-Hiding). *Let λ be a security parameter, LDP be a LDP mechanism, and*

P be a LDP-C. P is said to be IND-LDP-hiding if for any three-party PPT algorithm \mathcal{A} such that if the following value is positive:

$$\epsilon(\lambda) = \Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{IND-LDP-Hiding}}(\lambda) \right] - \Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{IND-LDP-Hiding}}(\lambda) \right]$$

then $\epsilon(\lambda)$ is negligible, where the IND-LDP-Hiding experiment is given in Figure 3.

While this definition provides an intuitive way to understand the IND-LDP-hiding, it does not provide a precise way to evaluate $\Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{IND-LDP-Hiding}}(\lambda) \right]$. To overcome this issue, we evaluate the winning probability of the algorithm OptiGuess in Theorem 1 (proven in Appendix A) in a more precise and usable way. Interestingly, it depends only on the distribution induced by LDP. Since the adversary's response is replaced by OptiGuess in $\text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{IND-LDP-Hiding}}(\lambda)$, the probability that the experiment returns 1 is the winning probability of OptiGuess (Figure 3), weighted by the probabilities of the adversary's choices for the (m_0, m_1) .

Theorem 1. *For any tuple $(\lambda, \text{LDP}, m_0, m_1)$ where the set of messages of the LDP mechanism is denoted by \mathcal{M} and $(m_0, m_1) \in \mathcal{M}$, we have:*

$$\Pr \left[\begin{array}{l} b \xleftarrow{\$} \{0, 1\}; \hat{m} \leftarrow \text{LDP}(m_b); \\ b_* \leftarrow \text{OptiGuess}(\lambda, \text{LDP}, \hat{m}, m_0, m_1) : b = b_* \end{array} \right] = \frac{1}{2} \sum_{\hat{m} \in \mathcal{M}} \max \left(\begin{array}{l} \Pr[\hat{m} \leftarrow \text{LDP}(m_0)], \\ \Pr[\hat{m} \leftarrow \text{LDP}(m_1)] \end{array} \right).$$

The following theorem shows a relation between our hiding properties.

Theorem 2. *A LDP-C that is both hiding and ROS-LDP-hiding is IND-LDP-hiding.*

Intuitively, the ROS-LDP-hiding property ensures that observing an anonymized message from the real LDP mechanism with a simulated proof is similar, from the adversary's point of view, to observing a message and a proof returned by the algorithm OpenLDP. The only additional information on m_b that the adversary knows is therefore the commitment c , however the hiding property ensures that c gives no information on m_b that can be significantly exploited by a PPT adversary. Finally, the best strategy left to the adversary is to try to guess m_b using only the anonymized message \hat{m} , as in the case of $b = 1$.

In the following, we provide a sketch of this proof, giving the sequence of games [27], but omitting the reductions between the games. The full proof is available in the appendix this paper.

Proof sketch. We use the following sequence of games:

Game G_0 : This game is the same as the IND-LDP-Hiding experiment in the case where $b = 0$.

Game G_1 : This game is the same as the game G_0 except that the challenger replaces the zero-knowledge proof $\hat{\pi}$ by a simulated proof, we have $\Pr [G_0 \text{ returns } 1] = \Pr [G_1 \text{ returns } 1]$.

Game G_2 : This game is the same as the game G_1 except

that \hat{m}_0 is obtained by computing $\text{LDP}(m_{b'})$. By reduction we can show that:

$$\begin{aligned} & |\Pr[G_1 \text{ returns } 1] - \Pr[G_2 \text{ returns } 1]| \\ & \leq \epsilon_{\text{ROS-LDP-Hiding}}(\lambda), \end{aligned}$$

where $\epsilon_{\text{ROS-LDP-Hiding}}$ is the advantage on the ROS-LDP-Hiding experiment.

Game G_3 : This game is the same as the game G_2 except that (k, c, π_*) is obtained by picking a random $m \xleftarrow{\$} \mathcal{M}$ and by computing $(k, c, \pi_*) \leftarrow \text{Commit}(m, \theta)$. By reduction we can show that:

$$|\Pr[G_2 \text{ returns } 1] - \Pr[G_3 \text{ returns } 1]| \leq \epsilon_{\text{Hiding}}(\lambda),$$

where ϵ_{Hiding} is the advantage on the Hiding experiment. In G_3 , the only information the adversary knows about $m_{b'}$ is $\text{LDP}(m_{b'})$, so their best strategy for guessing b' is to apply the algorithm $\text{OptiGuess}(\lambda, \text{LDP}, \hat{m}_1, m_0, m_1)$, and in this case, the probability that G_3 returns 1 is the same as the probability that the IND-LDP-Hiding experiment returns 1 in the case $b = 1$. It follows that the adversary's advantage $\epsilon(\lambda)$ over the experiment IND-LDP-Hiding is bounded by $\epsilon_{\text{ROS-LDP-Hiding}}(\lambda) + \epsilon_{\text{Hiding}}(\lambda)$. \square

The IND-LDP-hiding property matches better the security concept we want to formalize, however, the ROS-LDP-hiding property appears to be easier to prove and simplifies the proofs for the IND-LDP-hiding property. It is also stronger under the hypothesis that the scheme is hiding.

3.2.2. Binding. First of all, we define Binding in the classical way: an adversary is unable to open a commitment in two different ways (revealing two different messages) using the algorithm Open.

Definition 9 (Binding). Let λ be a security parameter, LDP be a LDP mechanism, and P be a LDP-C. P is said to be binding if for any PPT algorithm \mathcal{A} , there exists a negligible function ϵ s.t.:

$$\Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}}^{\text{Binding}}(\lambda) \right] \leq \epsilon(\lambda),$$

where the Binding experiment is given in Figure 3.

In the same way, we define the LDP-Binding experiment for the OpenLDP algorithm. LDP-Binding security is achieved when no PPT adversary is able to open a commitment on two different messages using the same seed on the algorithm OpenLDP.

Definition 10 (LDP-Binding). Let λ be a security parameter, LDP be a LDP mechanism, and P be a LDP-C. P is said to be LDP-binding if for any PPT algorithm \mathcal{A} , there exists a negligible function ϵ s.t.:

$$\Pr \left[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}}^{\text{LDP-Binding}}(\lambda) \right] \leq \epsilon(\lambda),$$

where the LDP-Binding experiment is given in Figure 3.

When the seed for algorithm OpenLDP is chosen randomly, the message opened should follow the same distribution as if it had been generated honestly using the LDP mechanism on the original committed message. We stress that in this case the same commitment must open

with different values depending on the seed, since the LDP mechanism is probabilistic by nature. We model this property by the experiment Prob-LDP-Binding, where an adversary commits a message m , receives a random seed $\hat{\theta}$, and opens the commitment with $\hat{\theta}$ using the OpenLDP algorithm. Their aim is to return an opened message \hat{m} chosen in a distribution which differs significantly from the distribution generated by the use of the LDP mechanism on m , while guaranteeing that the commitment is correct (i.e. the proof π_* is correct), that m is indeed the committed message (i.e. the proof π is correct), and that \hat{m} is indeed the message opened by the OpenLDP algorithm using $\hat{\theta}$ as seed (i.e. the proof $\hat{\pi}$ is correct). The experiment is parameterized by a bit b . When $b = 0$ and the commitment has been correctly generated and opened by the adversary, the experiment returns the anonymized value opened by the adversary. Otherwise, the experiment itself anonymizes the adversary's message and returns it. Probabilistic-LDP-binding security is achieved when, for any adversary, the message returned follows the same probability distribution, whether $b = 0$ or 1. In other words, the adversary cannot bias the LDP mechanism when it opens the commitment with OpenLDP on a random seed.

Definition 11 (Probabilistic-LDP-Binding). Let λ be a security parameter, LDP be a LDP mechanism, and P be a LDP-C. P is said to be probabilistic-LDP-binding if for any PPT algorithm \mathcal{A} and any message \hat{m} , there exists a negligible function ϵ s.t.:

$$\begin{aligned} & \left| \Pr \left[\hat{m} \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{Prob-LDP-Binding}}(\lambda) \right] - \right. \\ & \left. \Pr \left[\hat{m} \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{Prob-LDP-Binding}}(\lambda) \right] \right| \leq \epsilon(\lambda), \end{aligned}$$

the Prob-LDP-Binding experiment being given in Figure 3.

4. LDP Commitment Schemes

In this section, we present an efficient LDP-C scheme for the generalized randomized response mechanism.

4.1. Privacy Parameter for the LDP Mechanism

This LDP mechanism is parameterized by a pair (n_1, n_2) . Given a message $m \in \mathcal{M}$, the mechanism returns $\hat{m} = m$ with probability $1/n_1$, otherwise it draws $\hat{m} \xleftarrow{\$} \mathbb{Z}_{n_2}$ and returns \hat{m} .

Definition 12 (Generalized Randomized Response Mechanism). The generalized randomized response mechanism parameterized by (n_1, n_2) is the following locally differentially private algorithm:

$\text{LDP}(m)$: Picks $x \xleftarrow{\$} \mathbb{Z}_{n_1}$, if $x = 0$, then sets $\hat{m} \leftarrow m$, else picks $\hat{m} \xleftarrow{\$} \mathbb{Z}_{n_2}$, finally returns \hat{m} .

We now want to determine how to set the values of n_1 and n_2 given the privacy parameter ϵ and the probabilities

<p>$\text{Exp}_{P,\text{LDP},\mathcal{A},b}^{\text{Hiding}}(\lambda)$: set $\leftarrow \text{Setup}(\lambda, \text{LDP})$ $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$ $\theta \xleftarrow{\\$} \Theta$ $(k, c, \pi_*) \leftarrow \text{Commit}(m_b, \theta)$ $b_* \leftarrow \mathcal{A}_2(c, \pi_*)$ return $b_* = b$</p> <p>$\text{Exp}_{P,\text{LDP},\mathcal{A},b}^{\text{ROS-LDP-Hiding}}(\lambda)$: set $\leftarrow \text{Setup}(\lambda, \text{LDP})$ $m \leftarrow \mathcal{A}_1(\text{set})$ $\theta \xleftarrow{\\$} \Theta$ $(k, c, \pi_*) \leftarrow \text{Commit}(m, \theta)$ $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$ $(\hat{m}_0, \hat{\pi}_0) \leftarrow \text{OpenLDP}(k, c, \hat{\theta})$ $\hat{m}_1 \leftarrow \text{LDP}(m)$ $\hat{\pi}_1 \leftarrow \text{Sim}(\hat{m}_1, c)$ $b_* \leftarrow \mathcal{A}_3(\hat{m}_b, \hat{\pi}_b)$ return $b_* = b$</p> <p>$\text{Exp}_{P,\text{LDP},\mathcal{A},b}^{\text{IND-LDP-Hiding}}(\lambda)$: set $\leftarrow \text{Setup}(\lambda, \text{LDP})$ $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$ $b' \xleftarrow{\\$} \{0, 1\}$ $\theta \xleftarrow{\\$} \Theta$ $(k, c, \pi_*) \leftarrow \text{Commit}(m_{b'}, \theta)$ $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$ $(\hat{m}_0, \hat{\pi}) \leftarrow \text{OpenLDP}(k, c, \hat{\theta})$ $\hat{m}_1 \leftarrow \text{LDP}(m_{b'})$ if $b = 0$, then $b_* \leftarrow \mathcal{A}_3(\hat{m}_0, \hat{\pi})$ else $b_* \leftarrow \text{OptiGuess}(\lambda, \text{LDP}, \hat{m}_1, m_0, m_1)$ return $b_* = b'$</p>	<p>$\text{Exp}_{P,\text{LDP},\mathcal{A}}^{\text{Binding}}(\lambda)$: set $\leftarrow \text{Setup}(\lambda, \text{LDP})$ $(m_0, m_1, \pi_0, \pi_1, c) \leftarrow \mathcal{A}(\text{set})$ return $(m_0 \neq m_1 \wedge \text{VerOpen}(c, m_0, \pi_0) \wedge \text{VerOpen}(c, m_1, \pi_1))$</p> <p>$\text{Exp}_{P,\text{LDP},\mathcal{A}}^{\text{LDP-Binding}}(\lambda)$: set $\leftarrow \text{Setup}(\lambda, \text{LDP})$ $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta}) \leftarrow \mathcal{A}(\text{set})$ return $(\hat{m}_0 \neq \hat{m}_1 \wedge \text{VerOpenLDP}(c, \hat{m}_0, \hat{\pi}_0, \hat{\theta}) \wedge \text{VerOpenLDP}(c, \hat{m}_1, \hat{\pi}_1, \hat{\theta}) \wedge \text{VerCommit}(c, \pi_*))$</p> <p>$\text{Exp}_{P,\text{LDP},\mathcal{A},b}^{\text{Prob-LDP-Binding}}(\lambda)$: set $\xleftarrow{\\$} \text{Setup}(\lambda, \text{LDP})$ $(c, \pi_*, m, \pi) \leftarrow \mathcal{A}_1(\text{set})$ $\hat{\theta} \xleftarrow{\\$} \Theta$ $(\hat{m}_0, \hat{\pi}) \leftarrow \mathcal{A}_2(\hat{\theta})$ $\hat{m}_1 \leftarrow \text{LDP}(m)$ if $(\text{VerCommit}(c, \pi_*) \wedge \text{VerOpen}(c, m, \pi) \wedge \text{VerOpenLDP}(c, \hat{m}_0, \hat{\pi}, \hat{\theta}) = 1) \wedge (b = 0)$, then return \hat{m}_0, else return \hat{m}_1</p> <p>$\text{OptiGuess}(\lambda, \text{LDP}, \hat{m}, m_0, m_1)$: $b_* \xleftarrow{\\$} \{0, 1\}$ if $\Pr[\hat{m} \leftarrow \text{LDP}(m_0)] > \Pr[\hat{m} \leftarrow \text{LDP}(m_1)]$ then return 0 if $\Pr[\hat{m} \leftarrow \text{LDP}(m_0)] < \Pr[\hat{m} \leftarrow \text{LDP}(m_1)]$ then return 1 if $\Pr[\hat{m} \leftarrow \text{LDP}(m_0)] = \Pr[\hat{m} \leftarrow \text{LDP}(m_1)]$ then return b_*</p>
---	--

Figure 3. Security experiments for LDP-C.

of obtaining the correct (resp. an erroneous) value. We have, for any $\hat{m} \neq m$ in \mathbb{Z}_{n_2} :

$$\begin{aligned} \Pr[m \leftarrow \text{LDP}(m)] &= \frac{1}{n_1} + \frac{n_1 - 1}{n_1} \cdot \frac{1}{n_2} \\ &= \frac{n_1 + n_2 - 1}{n_1 \cdot n_2} \quad (\text{Eq.1}) \\ \Pr[\hat{m} \leftarrow \text{LDP}(m)] &= \frac{n_1 - 1}{n_1} \cdot \frac{1}{n_2} = \frac{n_1 - 1}{n_1 \cdot n_2} \end{aligned}$$

As LDP implements an optimal staircase mechanism (Section 2), we have the following constraint on the ratio:

$$\left| \ln \frac{\Pr[m \leftarrow \text{LDP}(m)]}{\Pr[\hat{m} \leftarrow \text{LDP}(m)]} \right| = \epsilon$$

so that we can express a relationship between n_1 , n_2 and ϵ :

$$n_1 = \frac{n_2 + \exp(\epsilon) - 1}{\exp(\epsilon) - 1}.$$

The parameters are chosen in the following manner. First of all, ϵ is fixed by the user, medical practitioner or medical device producing the value. Recall that the greater

the value of ϵ , the lesser the privacy of the mechanism, since the probability of yielding the same outcome as the input value will be higher. In principle, what a user is interested in fixing next, is the actual probability of outputting the correct value, *i.e.* $\Pr[m \leftarrow \text{LDP}(m)]$. With this parameter fixed, using Eq.1 we can compute:

$$n_1 = \frac{1}{\Pr[m \leftarrow \text{LDP}(m)]} \cdot \frac{\exp(\epsilon)}{\exp(\epsilon) - 1}$$

and deduce n_2 . If n_1 is an integer, we can choose to take the floor or ceiling value, depending if we want $\Pr[m \leftarrow \text{LDP}(m)]$ to be a target lower or upper bound.

Example. If we choose $\epsilon = \ln 3$ and $\Pr[m \leftarrow \text{LDP}(m)] = \frac{3}{4}$, this leads us to find $n_1 = n_2 = 2$ which is the classical Randomized Response algorithm [28].

4.2. Naive Solution

We start by giving a naive solution, which is intuitive and correct, but which quickly becomes ineffective when the settings grow larger. This solution uses a standard

cryptographic commitment scheme (Com, Ope), where $\text{Com}(m)$ returns a key k and a committed c , and $\text{Ope}(k, c)$ returns the committed message m . The general idea is to commit a vector containing the repetition of elements of \mathbb{Z}_{n_2} in such a way that the choice in the uniform distribution over the elements of this vector coincides with the distribution of values returned by the LDP mechanism on a given message. More precisely, given the message m , we define the following vector:

$$\begin{aligned} v &= (v_i)_{i=0}^{n_1 \cdot n_2} \\ &= \underbrace{(0, \dots, 0)}_{n_1 - 1}, \dots, \underbrace{m - 1, \dots, m - 1}_{n_1 - 1}, \underbrace{m, \dots, m}_{n_1 + n_2 - 1}, \\ &\quad \underbrace{m + 1, \dots, m + 1}_{n_1 - 1}, \dots, \underbrace{n_2 - 1, \dots, n_2 - 1}_{n_1 - 1}. \end{aligned}$$

To commit the message m with the algorithm $\text{Commit}(m, \theta)$ of our naive LDP-C, the user runs $(k_*, c_*) \leftarrow \text{Com}(m)$, parses θ as a bijective function $\theta : \mathbb{Z}_{n_1 \cdot n_2} \rightarrow \mathbb{Z}_{n_1 \cdot n_2}$, runs $(k_i, c_i) \leftarrow \text{Com}(v_{\theta(i)})$ for all $i \in \mathbb{Z}_{n_1 \cdot n_2}$, and returns the key $k = (k_*, (k_i)_{i \in \mathbb{Z}_{n_1 \cdot n_2}})$ and the commitment $c = (c_*, (c_i)_{i \in \mathbb{Z}_{n_1 \cdot n_2}})$. The user also proves using zero-knowledge proofs that each m is committed at least $n_1 - 1$ times in c and that the value committed in c_* is committed at least $n_1 + n_2 - 1$ times in c to generate the proof π_* . For instance, this proof can easily be built using Schnorr based proofs for discret logarithm relations [29] with the proofs of partial knowledge transformation [30] on Pedersen's commitments [31].

To open the commitment c with the algorithm $\text{Open}(k, c)$ (*i.e.*, without LDP), the user returns m together with a zero-knowledge proof π that $m = \text{Ope}(k_*, c_*)$ without revealing k_* . This proof can easily be built using Schnorr based proofs for discret logarithm relations on Pedersen's commitments.

Given a random seed $\hat{\theta}$ chosen from $\mathbb{Z}_{n_1 \cdot n_2}$, the user can also open the commitment c by applying the LDP mechanism with the algorithm $\text{OpenLDP}(k, c, \hat{\theta})$. To do this, the user returns the committed message \hat{m} in $c_{\hat{\theta}}$ and a proof $\hat{\pi}$ that $\hat{m} = \text{Ope}(k_{\hat{\theta}}, c_{\hat{\theta}})$ without revealing $k_{\hat{\theta}}$ (using, once again, Schnorr based proofs for discret logarithm relations on Pedersen's commitments).

The hiding and LDP-hiding properties derive from the hiding property of the commitment scheme used and from the fact that the proofs are zero-knowledge. The use of permutation ensures that the verifier cannot use the choice of $\hat{\theta}$ to determine whether the value \hat{m} is the original message or another message; \hat{m} provides as much information as a randomly chosen value v_i in v , and therefore as much information as a message that is actually anonymized with the generalized randomized response LDP mechanism. Binding and LDP-binding result directly from the binding property of the cryptographic commitment scheme used. The probabilistic-LDP-binding is ensured by the soundness of the zero-knowledge proofs, which guarantee that if $\hat{\theta}$ is indeed random, then the opening of the commitment $c_{\hat{\theta}}$ will follow the distribution of the LDP mechanism, since the c_i are commitments of the values of the permuted vector v .

The size of the signatures and the computational complexity of the commitments and verifications are linear in

$n_1 \cdot n_2$, which seems unsatisfactory, especially when we want to use large parameters.

4.3. Efficient Scheme with Logarithmic Commitments

We will now present a more efficient LDP-C scheme called Optimized Randomized Response Commitment (ORRC) whose complexity in computation time and in the size of commitments, proofs, and openings is at most $O(\max(\log_2(n_1), \log_2(n_2)))$. Our scheme uses parameters n_1 and n_2 which are powers of 2, so $n_1 = 2^{\ell_1}$ and $n_2 = 2^{\ell_2}$. In practical terms, this means that the messages are chosen from vectors of ℓ_2 bits. Moreover, for fixed ℓ_2 and ϵ , we will choose the smallest ℓ_1 such that $\ell_1 \geq \log_2 \left(\frac{2^{\ell_2} + \exp(\epsilon) - 1}{\exp(\epsilon) - 1} \right)$.

Definition 13 (ORRC scheme). *Let (ℓ_1, ℓ_2) be two integers. The optimized randomized response commitment scheme ORRC is a LDP-C scheme depending on the generalized randomized response LDP mechanism parameterized by $(2^{\ell_1}, 2^{\ell_2})$ consisting of algorithms (Setup, Commit, Open, OpenLDP, VerOpen, VerOpenLDP, VerCommit) described in the rest of this section.*

The setup algorithm generates a group of prime order and several elements of this group. More precisely, we generate pairs $(f_{i,0}, f_{i,1})$, $(g_{i,0}, g_{i,1})$, and $(h_{i,0}, h_{i,1})$, which will be used to commit bit strings. To do this, given an integer ℓ and a bit string $s \in \{0, 1\}^\ell$, and using the pairs $(g_{i,0}, g_{i,1})$, we choose a random element x , set $y = g^x$, and compute $S_i = g_{i,s[i]}$ for all i in $[\ell]$. The opening of the commitment consists in revealing x . In this way, we can simply prove, using proofs of equality of discrete logarithms, that the committed message is indeed an element of $\{0, 1\}^\ell$; it suffices to show, for all i , that $S_i = g_{i,0}$ or $S_i = g_{i,1}$.

Note that it is also possible to check the commitment quite efficiently by checking $\prod_{i \in [\ell]} S_i^x = \left(\prod_{i \in [\ell]} g_{i,s[i]} \right)^x$ and $y = g^x$, rather than by checking each of the S_i independently. This is because exponentiation in a group of prime order is much more expensive than multiplication, so it is much more efficient to have a constant number of exponentiations that do not depend on ℓ , even with a linear number of multiplications.

Setup(λ , LDP): generates a group $\mathbb{G} = \langle g \rangle$ of prime order p , for each $i \in [\ell_1]$ and $j \in \{0, 1\}$, picks $f_{i,j} \xleftarrow{\$} \mathbb{G}$, then for each $i \in [\ell_2]$ and each $j \in \{0, 1\}$, picks $g_{i,j}, h_{i,j} \xleftarrow{\$} \mathbb{G}$. It sets $\mathcal{M} \leftarrow \{0, 1\}^{\ell_2}$ and $\Theta \leftarrow \{0, 1\}^{\ell_1} \times \{0, 1\}^{\ell_2}$, and returns set $\leftarrow (\lambda, \mathbb{G}, p, (g_{i,j})_{\substack{i \in [\ell_1] \\ j \in \{0,1\}}}, (f_{i,j}, h_{i,j})_{\substack{i \in [\ell_2] \\ j \in \{0,1\}}}, \mathcal{M}, \Theta)$.

The idea behind our scheme is as follows: given a θ token, we parse θ into (s, t) . To commit m with ORRC, we commit s , m , and t in the respective commitments A_1 , A_2 and B (using the method described above). To open without LDP, we simply open A_2 . To open with the LDP mechanism, given a seed $\hat{\theta} \in \Theta$ parsed as (\hat{s}, \hat{t}) chosen by the verifier, we test whether $s = \hat{s}$ (which happens with a probability of $1/2^{\ell_1}$). If so, we open the actual message m from A_2 , otherwise we return the message $\hat{m} = t \oplus \hat{t}$ with proof that \hat{m} has been computed

correctly with the value t committed to B (we stress that if t or \hat{t} was indeed chosen randomly, the message \hat{m} will follow a uniform distribution over \mathcal{M}). In addition, we use a zero-knowledge proof to prove that the correct operation between the two possible has been performed, depending on whether $s = \hat{s}$ has been performed, and without revealing s .

Commit(m, θ): picks $x \in \mathbb{Z}_p^*$, sets $y = g^x$, and parses θ as (s, t) . For all $i \in \llbracket \ell_1 \rrbracket$, it sets $A_{(i,1)} \leftarrow g_{i,s[t]}^x$. For all $i \in \llbracket \ell_2 \rrbracket$, it sets $(A_{(i,2)}, B_{(i,0)}, B_{(i,1)}) \leftarrow (f_{i,m[t]}^x, h_{i,t[t]}^x, h_{i,1 \oplus t[t]}^x)$. It then computes:

$$\left. \begin{array}{l} \pi_{A_1} \leftarrow \text{NIP} \left\{ x : \bigwedge_{i=1}^{\ell_1} \left(\begin{array}{l} y = g^x \\ \bigvee_{j=0}^1 A_{(i,1)} = g_{i,j}^x \end{array} \right) \right\} \\ \pi_{A_2} \leftarrow \text{NIP} \left\{ x : \bigwedge_{i=1}^{\ell_2} \left(\begin{array}{l} y = g^x \\ \bigvee_{j=0}^1 A_{(i,2)} = f_{i,j}^x \end{array} \right) \right\} \\ \pi_B \leftarrow \text{NIP} \left\{ x : \bigwedge_{i=1}^{\ell_2} \left(\begin{array}{l} y = g^x \\ \bigwedge_{j=0}^1 B_{(i,j)} = h_{i,j}^x \\ \bigvee_{j=0}^1 B_{(i,j)} = h_{i,1-j}^x \end{array} \right) \right\} \end{array} \right\}.$$

It sets $k \leftarrow x, c \leftarrow (y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket}), \pi_* \leftarrow (\pi_{A_1}, \pi_{A_2}, \pi_B)$. Finally, it returns (k, c, π_*) .

VerCommit(c, π_*): parses c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$ and π_* as $(\pi_{A_1}, \pi_{A_2}, \pi_B)$, verifies the proofs in π_* , and returns 1 if the proof is valid, 0 otherwise.

Note that the $B_{(i,1)}$ commit \bar{t} , we will see later that these elements will be useful for showing that in the case $s \neq \hat{s}$, $\hat{m} = t \oplus \hat{t}$. The opening algorithm with no LDP consists of opening the $A_{(i,2)}$ in which m is committed, using the method given above.

Open(k, c): parses k as x and c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$. For each $i \in \llbracket \ell_2 \rrbracket$, it sets $m[i] = b$ iff $A_{(i,2)} = f_{i,b}^x$ where $b \in \{0, 1\}$. It computes $A_2 = \prod_{i=1}^{\ell_2} A_{(i,2)}$ and $\alpha_2 = \prod_{i=1}^{\ell_2} f_{i,m[i]}^x$. It computes the proof $\pi \leftarrow \text{NIP} \{x : y = g^x \wedge A_2 = \alpha_2^x\}$. Finally, it returns m and π .

VerOpen(c, m, π): parses c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$, computes A_2 and α_2 as in the algorithm Open and verifies the proof π .

As we already mentioned, to open the commitment with the LDP mechanism on the message \hat{m} and the seed $\hat{\theta} = (\hat{s}, \hat{t})$, we show that if s , the value committed to $A_{(i,1)}$, is equal to \hat{s} , then the value committed to $A_{(i,2)}$ is \hat{m} , otherwise we show that $\hat{m} = t \oplus \hat{t}$ where t is the value committed to $B_{(i,0)}$. We will now focus on the second case. Recall that for all i , $B_{(i,0)}$ commits $t[i]$ and $B_{(i,1)}$ commits $1 \oplus t[i]$. We remark that:

$$\begin{aligned} h_{i,t[\hat{i}]}^x &= \begin{cases} h_{i,t[\hat{i}]}^x & \text{if } t[\hat{i}] = \hat{t}[\hat{i}], \\ h_{i,1 \oplus t[\hat{i}]}^x & \text{else } (t[\hat{i}] \neq \hat{t}[\hat{i}]). \end{cases} \\ &= \begin{cases} B_{(i,0)} & \text{if } t[\hat{i}] = \hat{t}[\hat{i}], \\ B_{(i,1)} & \text{else } . \end{cases} \\ &= B_{(i,(t \oplus \hat{t})[\hat{i}])} = B_{(i,\hat{m}[\hat{i}])}. \end{aligned}$$

Therefore, to show that $m = t \oplus \hat{t}$, it is sufficient to prove that $(\prod_{i \in \llbracket \ell_2 \rrbracket} h_{i,\hat{t}[\hat{i}]})^x = \prod_{i \in \llbracket \ell_2 \rrbracket} B_{(i,\hat{m}[\hat{i}])}$ and $y = g^x$. This results in the following opening algorithm.

OpenLDP($k, c, \hat{\theta}$): parse k as x , c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$ and $\hat{\theta}$ as (\hat{s}, \hat{t}) . It sets:

$$A_1 = \prod_{i=1}^{\ell_1} A_{(i,1)}; \quad A_2 = \prod_{i=1}^{\ell_2} A_{(i,2)}; \quad \alpha_1 = \prod_{i=1}^{\ell_1} g_{i,\hat{s}[\hat{i}]};$$

If $A_1 = \alpha_1^x$ it sets $\hat{m} \leftarrow m$, else $\hat{m} \leftarrow t \oplus \hat{t}$. It sets:

$$\alpha_2 = \prod_{i=1}^{\ell_2} f_{i,\hat{m}[\hat{i}]}; \quad \beta = \prod_{i=1}^{\ell_2} B_{(i,\hat{m}[\hat{i}])}; \quad \gamma = \prod_{i=1}^{\ell_2} h_{i,\hat{t}[\hat{i}]},$$

then it generates the following proof:

$$\hat{\pi} \leftarrow \text{NIP} \left\{ x : \begin{array}{l} (y = g^x \wedge A_1 = \alpha_1^x \wedge A_2 = \alpha_2^x) \\ \vee (y = g^x \wedge A_1 \neq \alpha_1^x \wedge \beta = \gamma^x) \end{array} \right\}.$$

Finally, it returns $(\hat{m}, \hat{\pi})$.

VerOpenLDP($c, \hat{m}, \hat{\pi}, \hat{\theta}$): parses $\hat{\theta}$ as (\hat{s}, \hat{t}) and c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$, then computes $A_1, A_2, \alpha_1, \alpha_2, \beta$, and γ as in the algorithm OpenLDP. It verifies the proof $\hat{\pi}$, and returns 1 if the proof is valid, 0 otherwise.

4.4. Security Analysis

Theorem 3. *The ORRC scheme instantiated with extractable and zero-knowledge proofs and a group where the DDH assumption holds is hiding, LDP-hiding, binding, LDP-binding and probabilistic-LDP-binding.*

In the following, we provide sketches of the security proofs, giving the sequence of games [27] used for the proof, but omitting the reductions between the games. The full proofs are available in the appendix of this paper.

The hiding proofs requires the two following assumptions in the chosen group $\mathbb{G} = \langle g \rangle$ of prime order p :

Left-or-Right-DDH (LOR-DDH) the advantage of any PPT algorithm \mathcal{A} that receives random elements g^x, g^{y_0}, g^{y_1} , and g^{xy_b} for a random bit b and that tries to guess b is bounded by a negligible function $\epsilon_{\text{LOR-DDH}}(\lambda)$.

Left-or-Right-DDH-2 (LOR-DDH-2) the advantage of any PPT algorithm \mathcal{A} that receives random elements g^x, g^{y_0}, g^{y_1} , and g^{xy_b} and $g^{xy_{1-b}}$ for a random bit b and that tries to guess b is bounded by a negligible function $\epsilon_{\text{LOR-DDH-2}}(\lambda)$.

We prove in Appendix B that these two assumptions reduce to DDH (Lemmas 2 and 3).

Proof sketch that ORRC is hiding. We use the following sequence of games:

Game G_0 : This game is the same as the Hiding experiment in the case where $b = 0$.

Game G_1 : This game is the same as the game G_0 except that the challenger replaces the zero-knowledge proof π_* by a simulated proof, we have:

$$\Pr[\mathcal{A} \text{ returns 1 in } G_0] = \Pr[\mathcal{A} \text{ returns 1 in } G_1].$$

Game G_2 : This game is the same as the Hiding experiment in the case where $b = 1$.

In the game G_1 (resp. G_2), the challenger commits the message m_0 (resp. m_1), so the i -th bit of m_0 (resp. m_1) is contained in the part $A_{(i,2)}$ of c . To prove indistinguishability between G_1 and G_2 , we consider the following hybrid argument [32]:

Game $G_{1,k}$ (for all $k \in \llbracket \ell_2 \rrbracket$): We define $G_{1,0}$ as G_1 and $\overline{G_{1,\ell_2}}$ as $\overline{G_2}$. This game is the same as the game $G_{1,k-1}$ except that the challenger replaces $A_{(k,2)} = f_{k,m_0}^x$ by f_{k,m_1}^x in the commitment. By reduction we can show that for all $k \in \llbracket \ell_2 \rrbracket$:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1}] - \Pr[\mathcal{A} \text{ returns 1 in } G_{1,k}]| \\ & \leq \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

This leads to the advantage $\epsilon(\lambda) = \ell_2 \cdot \epsilon_{\text{LOR-DDH}}(\lambda)$ for the hiding property, which concludes the proof. \square

Proof sketch that ORRC is ROS-LDP-hiding. We define the algorithm LDP for some value $\hat{s} \in \llbracket \ell_1 \rrbracket$ and the simulator as follows:

LDP(m): picks $(s, t) \stackrel{\$}{\leftarrow} \{0, 1\}^{\ell_1} \times \{0, 1\}^{\ell_2}$, if $s = 0$, then sets $\hat{m} = m$, else, sets $\hat{m} = t$ and returns \hat{m} .
 Sim(\hat{m}): uses the zero-knowledge proof simulator to produce $\hat{\pi}$.

We use the following sequence of games:

Game G_0 : This game is the same as the ROS-LDP-Hiding experiment in the case $b = 0$.

Game G_1 : This game is the same as the game G_0 except that the challenger replaces the proofs $\hat{\pi}$ and π_* by simulated proofs. The loss of advantage between the games is the same as between G_0 and G_1 in the previous proof.

From the game G_1 , we parse the θ used to produce c as (s_0, t_0) . We note that if $s_0 = \hat{s}$, then $\hat{m} = m$, else $\hat{m} = t_0 \oplus t$. In what follows, we will gradually replace (s_0, t_0) (bit by bit) by other values (s_1, t_1) chosen at random in the commitment c , but will continue to compute \hat{m} with (s_0, t_0) . This leads us to the ROS-LDP-Hiding experiment in the case $b = 1$.

Game G_2 : This game is the same as the game G_1 except that for all $i \in \llbracket \ell_1 \rrbracket$, the challenger replaces $A_{i,1} = g_{i,s_0}^x$ by g_{i,s_1}^x . To prove indistinguishability between G_1 and G_2 , we consider the following hybrid argument [32]:

Game $G_{1,k}$ (for all $k \in \llbracket \ell_1 \rrbracket$): We define $G_{1,0}$ as G_1 and $\overline{G_{1,\ell_1}}$ as $\overline{G_2}$. This game is the same as the game $G_{1,k-1}$ except that the challenger replaces $A_{(k,1)} = g_{k,s_0}^x$ by g_{k,s_1}^x . By reduction we can show that for all $k \in \llbracket \ell_1 \rrbracket$:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1}] - \Pr[\mathcal{A} \text{ returns 1 in } G_{1,k}]| \\ & \leq \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

Game G_3 : This game is the same as the game $\overline{G_2}$ except that for all $i \in \llbracket \ell_2 \rrbracket$, the challenger replaces $(B_{(i,0)}, B_{(i,1)}) = (h_{i,t_0}^x, h_{i,1 \oplus t_0}^x)$ by $(h_{i,t_1}^x, h_{i,1 \oplus t_1}^x)$. To prove indistinguishability between G_2 and G_3 , we consider the following hybrid argument [32]:

Game $G_{2,k}$ (for all $k \in \llbracket \ell_2 \rrbracket$): We define $G_{2,0}$ as G_2 and $\overline{G_{2,\ell_2}}$ as $\overline{G_3}$. This game is the same as the game $G_{2,k-1}$ except that the challenger replaces $(B_{(k,0)}, B_{(k,1)}) =$

$(h_{k,t_0}^x, h_{k,1 \oplus t_0}^x)$ by $(h_{k,t_1}^x, h_{k,1 \oplus t_1}^x)$. By reduction we can show that for all $k \in \llbracket \ell_2 \rrbracket$:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1}] - \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k}]| \\ & \leq \epsilon_{\text{LOR-DDH-2}}(\lambda). \end{aligned}$$

This leads to the advantage $\epsilon(\lambda) = \ell_1 \cdot \epsilon_{\text{LOR-DDH}}(\lambda) + \ell_2 \cdot \epsilon_{\text{LOR-DDH-2}}(\lambda)$ for the hiding property, which concludes the proof. \square

According to Theorem 2, ORRC is IND-LDP-hiding because it is hiding and ROS-LDP-hiding.

The hiding proofs requires the following assumptions in the chosen group $\mathbb{G} = \langle g \rangle$ of prime order p :

ℓ -product-collision-resistance (ℓ -col) the probability of any PPT algorithm \mathcal{A} that receives random elements $(g_{(i,j)})_{i \in \llbracket \ell \rrbracket; j \in \{0,1\}}$ to find some (x_1, x_2) such that $x_1 \neq x_2$ and $\prod_{i=1}^{\ell} g_{(i,x_1[i])} = \prod_{i=1}^{\ell} g_{(i,x_2[i])}$ is bounded by a negligible function $\epsilon_{\ell\text{-col}}(\lambda)$.

We prove in Appendix B that this assumption reduces to the discrete logarithm assumption (Lemma 4).

Proof sketch that ORRC is binding. We use the following sequence of games:

Game G_0 : This game is the same as the Binding experiment.

Game G_1 : This game is the same as the game G_0 except that the challenger uses the extractors on the zero-knowledge proofs π_0, π_1 outputted by the adversary, and aborts and returns 0 on the event $F_1 =$ ‘‘An extractor fails on at least one proof’’. The loss of advantage between the games is the same as between G_0 and G_1 in the previous proof. We emphasize that if \mathcal{A} wins its game, then the witnesses x of all proofs are correctly extracted and are the same according to the proofs structure. We have:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq \Pr[F_1] \leq 2 \cdot \epsilon_{\text{ext}}(\lambda). \end{aligned}$$

Game G_2 : This game is the same as the game G_0 except that aborts and returns 0 on the event $F_2 =$ ‘‘ \mathcal{A} outputs $(m_0, m_1, \pi_0, \pi_1, c)$ such that $\prod_{i=1}^{\ell_2} f_{i,m_0[i]} = \prod_{i=1}^{\ell_2} f_{i,m_1[i]}$ and $m_0 \neq m_1$ ’’, we have:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ & \leq \Pr[F_2]. \end{aligned}$$

By reduction we can show that $\Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda)$. If F_1 and F_2 does not happen, then the values $A_{(i,2)}$ are correct and there is no product collision for two different messages, so $m_0 = m_1$ and so $\Pr[\mathcal{A} \text{ wins the game } G_2] = 0$. This leads to the advantage $\epsilon(\lambda) = 2 \cdot \epsilon_{\text{ext}}(\lambda) + \epsilon_{\ell_2\text{-col}}(\lambda)$ for the binding property, which concludes the proof. \square

Proof sketch that ORRC is LDP-binding. We use the following sequence of games:

Game G_0 : This game is the same as the LDP-Binding experiment.

Game G_1 : This game hop is similar to the hop between $\overline{G_0}$ and $\overline{G_1}$ in the previous proof except that we use the extractor on the proofs $\hat{\pi}_0, \hat{\pi}_1$ and $\pi_* = (\pi_{A_1}, \pi_{A_2}, \pi_B)$.

Game G_2 : This game is the same as the game G_1 except that aborts and returns 0 on the event $F_2 =$ ‘‘ \mathcal{A} returns

(\hat{m}_0, \hat{m}_1) such that $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]} = \prod_{i=1}^{\ell_2} f_{i, \hat{m}_1[i]}$ and $\hat{m}_0 \neq \hat{m}_1$. we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \leq \Pr[F_2].$$

By reduction we can show that $\Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda)$. Note that in the case where $\hat{s} \neq s$, we have $h_{i, \hat{t}[i]}^x = h_{i, \hat{m}[i] \oplus t[i]}^x$.

Game G_3 : This game is the same as the game G_2 except that aborts and returns 0 on the event $F_3 = \text{“}\mathcal{A} \text{ returns } (\hat{m}_0, \hat{m}_1) \text{ such that } \prod_{i=1}^{\ell_2} h_{i, \hat{m}_0[i] \oplus t[i]} = \prod_{i=1}^{\ell_2} h_{i, \hat{m}_1[i] \oplus t[i]} \text{ and } \hat{m}_0 \neq \hat{m}_1 \text{ and } s \neq \hat{s}\text{”}$, we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_2] - \Pr[\mathcal{A} \text{ wins the game } G_3]| \leq \Pr[F_3].$$

By reduction we can show that $\Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda)$.

Game G_4 : This game is the same as the game G_3 except that aborts and returns 0 on the event $F_4 = \text{“}\mathcal{A} \text{ returns } (\hat{m}_0, \hat{m}_1) \text{ such that } \prod_{i=1}^{\ell_2} B_{(i, \hat{m}_0[i])} = \prod_{i=1}^{\ell_2} B_{(i, \hat{m}_1[i])} \text{ and } \hat{m}_0 \neq \hat{m}_1 \text{ and } s \neq \hat{s}\text{”}$, we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_3] - \Pr[\mathcal{A} \text{ wins the game } G_4]| \leq \Pr[F_4].$$

By reduction we can show that $\Pr[F_4] \leq \epsilon_{\ell_2\text{-col}}(\lambda)$. If F_1, F_2, F_3 , and F_4 does not happen, then the commitment is well formed and there is no product collision for two different messages on the α_2, β , and γ , which implies the uniqueness of the open message for (\hat{s}, \hat{t}) , so $m_0 = m_1$ and so $\Pr[\mathcal{A} \text{ wins the game } G_4] = 0$. This leads to the advantage $\epsilon(\lambda) = 5 \cdot \epsilon_{\text{ext}}(\lambda) + 3 \cdot \epsilon_{\ell_2\text{-col}}(\lambda)$ for the LDP-binding property, which concludes the proof. \square

Proof sketch that ORRC is probabilistic-LDP-binding.

We use the following sequence of games:

Game G_0 : This game is the same as the Prob-LDP-Binding experiment in the case where $b = 0$.

Game G_1 : This game hop is similar to the hop between \bar{G}_0 and \bar{G}_1 in the previous proof.

Game G_2 : This game is the same as the game G_1 except that the challenger aborts and returns $\hat{m} \xleftarrow{\$} \mathcal{M}$ on the event $F_2 = \text{“}\mathcal{A}_1 \text{ returns } (c, \pi_*, m, \pi) \text{ and } \mathcal{A}_2 \text{ returns } (\hat{m}_0, \hat{\pi}) \text{ such that } \prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]} = \prod_{i=1}^{\ell_2} f_{i, m[i]} \text{ and } \hat{m}_0 \neq m\text{”}$, we have that $\forall \hat{m} \in \mathcal{M}$:

$$|\Pr[\hat{m} \leftarrow G_1] - \Pr[\hat{m} \leftarrow G_2]| \leq \Pr[F_2].$$

By reduction we can show that $\Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda)$. Note that in the case where $\hat{s} \neq s$, we have $\hat{m} = t \oplus \hat{t}$, which implies $h_{i, \hat{t}[i]}^x = h_{i, \hat{m}[i] \oplus t[i]}^x$ and $B_{(i, \hat{m}[i])} = h_{i, \hat{t}[i]}^x$.

Game G_3 : This game is the same as the game G_2 except that aborts and returns $\hat{m} \xleftarrow{\$} \mathcal{M}$ on the event $F_3 = \text{“}\mathcal{A}_1 \text{ returns } (c, \pi_*, m, \pi) \text{ and } \mathcal{A}_2 \text{ returns } (\hat{m}_0, \hat{\pi}) \text{ such that } \prod_{i=1}^{\ell_2} h_{i, \hat{m}_0[i] \oplus t[i]} = \prod_{i=1}^{\ell_2} h_{i, \hat{t}[i]} \text{ and } \hat{m}_0 \oplus t \neq \hat{t} \text{ and } s \neq \hat{s}\text{”}$, we have that, $\forall \hat{m} \in \mathcal{M}$:

$$|\Pr[\hat{m} \leftarrow G_3] - \Pr[\hat{m} \leftarrow G_4]| \leq \Pr[F_3].$$

By reduction we can show that $\Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda)$.

Game G_4 : This game is the same as the game G_3 except that aborts and returns $\hat{m} \xleftarrow{\$} \mathcal{M}$ on the event $F_4 = \text{“}\mathcal{A}_1$

returns (c, π_*, m, π) such that $\prod_{i=1}^{\ell_1} g_{i, s[i]} = \prod_{i=1}^{\ell_1} g_{i, \hat{s}[i]}$ and $s \neq \hat{s}\text{”}$, we have that, $\forall \hat{m} \in \mathcal{M}$:

$$|\Pr[\hat{m} \leftarrow G_3] - \Pr[\hat{m} \leftarrow G_4]| \leq \Pr[F_4].$$

By reduction we can show that $\Pr[F_4] \leq \epsilon_{\ell_1\text{-col}}(\lambda)$.

If F_1, F_2, F_3 , and F_4 does not happen, then the commitment is well formed and the only way to open the commitment is to return m if $s = \hat{s}$, and to return $t \oplus \hat{t}$ otherwise. We deduce that G_4 is the same as the Prob-LDP-Binding experiment in the case where $b = 1$. This leads to the advantage $\epsilon(\lambda) = 5 \cdot \epsilon_{\text{ext}}(\lambda) + \epsilon_{\ell_1\text{-col}}(\lambda) + 2 \cdot \epsilon_{\ell_2\text{-col}}(\lambda)$ for the probabilistic-LDP-binding property, which concludes the proof. \square

5. Extensions

In this section, we propose three extensions to our scheme. The first concerns the case where the same commitment can be opened several times using the LDP mechanism with different seeds. The second shows that it is possible to open the commitment with a LDP parameter ϵ larger (*i.e.* less privacy, more utility) than the one chosen at the time of the commitment without interacting with the user having committed the data. Finally, the last extension shows how to adapt our scheme to choose a parameter n_1 that is not a power of 2.

Extension for Multiple Openings. Our scheme supports only one opening using LDP (or several openings to non-colluding users). For instance, given two openings for the commitment of m giving the messages \hat{m} and \hat{m}' for the seeds $\hat{\theta} = (\hat{s}, \hat{t})$ and $\hat{\theta}' = (\hat{s}', \hat{t}')$ such that $\hat{\theta} \neq \hat{\theta}'$, if $\hat{m} = \hat{m}'$, we can deduce that $\hat{m} = \hat{m}' = m$, otherwise, if $\hat{m} \oplus \hat{m}' = \hat{t} \oplus \hat{t}'$, we can deduce with a high probability that $\hat{m} \neq m$ and $\hat{m}' \neq m$. Note that, although it may be possible to limit the leakage of information about the message, the problem is inherent in the primitive: an attacker that could try each of the seeds will always find the original message with certainty. The number of openings with LDP cannot therefore be variable and must be fixed at the time of signing.

We propose here a simple solution to allow a fixed number N of openings, at the cost of a linear factor in N on the size of the commitment and its generation time (note that it is in any case undesirable to open the same data several times with LDP because of the composition theorem [1] which shows that opening N times the same data with an ϵ -differentially private mechanism amounts to applying a $(N\epsilon)$ -differentially private mechanism).

The solution is simply to commit to the same message N times with ORRC, and to open a different commitment for each opening query. We denote $c = (c_i)_{i \in [N]}$ the commitment constructed in this way, with for each $i \in [N]$, $c_i = (y_i, (A_{(i,j,1)})_{j \in [l_1]}, (A_{(i,j,2)}, B_{(i,j,0)}, B_{(i,j,1)})_{j \in [l_2]})$, and $k = (k_i)_{i \in [N]}$ the corresponding opening secret key. In the case where the user having committed to the message is not considered to be honest, it is also necessary to ensure that it is indeed the same message that has been committed in each of the N ORRC commitments. This can easily be shown using the following proofs of knowledge of a

discrete logarithm for each (i, j) in $(\llbracket N \rrbracket \setminus \{1\}) \times j \in \llbracket \ell_2 \rrbracket$ where $k_i^* = k_i/k_1$:

$$\pi_{(i,j)} \leftarrow \text{NIP} \left\{ k_i^* : y_1^{k_i^*} = y_i \wedge A_{(1,j,2)}^{k_i^*} = A_{(i,j,2)} \right\}.$$

Combined with the other proofs generated during the commitment, this proof proves that $y_1 = g^{k_1}$ and $A_{(1,j,2)} = f_{(j,0)}^{k_1}$ and $y_i = g^{k_i}$ and $f_{(j,0)}^{k_i}$, or $y_1 = g^{k_1}$ and $A_{(1,j,2)} = f_{(j,1)}^{k_1}$ and $y_i = g^{k_i}$ and $f_{(j,1)}^{k_i}$, which proves that the j -th bit of the committed message is the same in the first and i -th commitments. Combining all these proofs, we have that all the committed messages are actually the same. We leave as an open problem the question of finding a way to allow multiple openings without a linear factor in time and size complexity for the generation of the commitment.

Opening with Larger ϵ . In the context of our application, where the user who commits the data and the user who reveals it are two different entities, it may be useful to decide at the time of opening which ϵ will be used. This would make it possible to decide on the ϵ according to the intended use of the data, which is not necessarily known in advance. We note that this is possible with our scheme by reducing the parameter $n_1 = 2^{\ell_1}$. Remember that the commitment uses a binary string s of ℓ_1 bits, so that each of its bits is committed to the $(A_{i,1})_{i \in \llbracket \ell_1 \rrbracket}$, and which is used to decide whether the opening with the LDP will return the value actually committed or a randomly chosen value (depending on whether $s = \hat{s}$ or not, where \hat{s} is also a string of ℓ_1 bits chosen by the verifier). If we need to use an LDP mechanism parameterized by $n'_1 = 2^{\ell'_1}$ with $\ell'_1 < \ell_1$, we simply choose a \hat{s} of ℓ'_1 bits and compare it with the binary string s' made up of the first ℓ'_1 bits of s , using the elements $(A_{i,1})_{i \in \llbracket \ell'_1 \rrbracket}$.

Fine-grained choice of n_1 . As it stands, our scheme requires n_1 to be a power of 2, which does not allow us to precisely choose the probability $1/n_1$ for the opener to open on the original value. To do this, we should set s to be the ℓ_1 -bit coding of a number chosen at random between 0 and $n_1 - 1$ where $n_1 < 2^{\ell_1}$, and ask the verifier that \hat{s} also be the ℓ_1 -bit coding of a number less than n_1 . On the other hand, it would be necessary to prove that the committed s has been chosen correctly, *i.e.* that $s < n_1$. Such a proof is constructed in [33] for another application. Note that this proof concerns commitments constructed in the same way as in our case and can be used without adaptation (for each bit, the two possible values correspond to two group generators, and the generator corresponding to the actual bit value is used to compute the exponentiation of a secret), and that the complexity in time and size of this proof is in $O(\ell_1)$, so its use does not affect the overall complexity in $O(\max(\log_2(n_1), \log_2(n_2)))$ of our construction.

6. Instantiation and Implementation

Instantiation of the NIP. All the NIP languages used in our scheme correspond to boolean relations of equality and inequality of discrete logarithms. These proofs can be instantiated as sigma protocols, where the prover sends a commitment, receives a challenge, and returns a response.

By using the hash (produced by a random oracle) of the statement and the commitment as a challenge, these proofs become non-interactive [34]. Such a proof for the equality of discrete logarithms is given in [29], and a proof for the inequality is given in [35]. To prove that several statements are true at the same time, it is sufficient to use the same challenge for all the proofs. To prove that a single statement is true among several, we can use the transformation given in [30]. The proofs produced in this way are linear in the number of statements (in size and time). These proofs are zero-knowledge and extractable, so we can use them to instantiate our construction.

Instantiation of the signature. Remember that in our application, the LDP commitment needs to be signed. The data is committed to by a user, and will be opened (with LDP) by a delegate, who must be able to convince a verifier that the data it receives is indeed the data committed by the user after the LDP mechanism has been applied. From a formal point of view, a signature is a triplet of algorithms $(\text{Gen}, \text{Sign}, \text{Ver})$ such that $\text{Gen}(\lambda)$ returns a private/public key pair (pk, sk) , $\text{Sign}(\text{sk}, m)$ returns a signature σ for message m , and $\text{Ver}(\text{pk}, m)$ decides whether a signature is valid or not. A signature is EUF-CMA if no polynomial adversary is able, given the public key pk and an oracle that produces signatures $\sigma \leftarrow \text{Sign}(\text{sk}, m)$ for chosen messages m , to forge a valid and fresh signature (*i.e.* one that has not been produced by the oracle). In our implementation, we use the well-known Schnorr's signature [36].

Implementation. In order to evaluate the efficiency of our LDP commitment scheme, we implemented the algorithm Setup, Commit, VerCommit, Open, VerOpen, OpenLDP and VerOpenLDP in Rust on a processor 11th Gen Intel® Core™ i7-1185G7 @ 3.00GHz × 8. The source code is available at [37].

We employed the prime order group Ristretto and we used the curve25519_dalek [38] library, with 255 bits secret keys. Additionally, we implemented a Schnorr Signature scheme constituted of the algorithms Gen, Sign and Ver allowing to sign the commitment produced. We measured the execution time of each algorithm over 1000 runs. The results are summarized in Table 1; They depend on the length of the parameters ℓ_1 and ℓ_2 thus for message space sizes $n_2 = 2^{\ell_2}$ ranging from 16 to 10^9 and ϵ around 0.70 (between 0.69 and 0.72). To improve our performance results, we evaluated the elapsed time with optimized features. We emphasize that our solution is particularly efficient for the verifier of openings, as we had suspected. The commitment correction verification takes a little longer, but it can be pre-computed before the commitment is opened, since it only concerns the commitment and not its opening.

7. Conclusion

In this paper, we proposed a security model and a proven scheme for a new primitive called LDP commitment. Our scheme uses standard cryptographic tools and is very efficient. We have implemented it in Rust to analyse its performance and show that it can be used in practice. By signing an LDP commitment, a user allows a delegate

Table 1. RUNNING TIME IN MILLISECONDS OF OUR ALGORITHMS.
THE RESULTS ARE AN AVERAGE OVER 1000 RUNS.

(ℓ_1, ℓ_2)	(4, 4)	(7, 7)	(20, 20)	(30, 30)
Setup	0.30	0.54	1.54	2.25
Commit	3.26	5.80	15.32	25.10
VerCommit	2.76	4.93	13.32	21.24
Open	0.30	0.51	1.10	1.71
VerOpen	0.13	0.14	0.14	0.15
OpenLDP	0.75	0.85	1.14	1.50
VerOpenLDP	0.51	0.52	0.51	0.56
Gen	0.03	0.03	0.03	0.03
Sign	0.12	0.18	0.39	0.61
Ver	0.15	0.20	0.44	0.62

to add noise before revealing their data to a recipient who will be convinced that they indeed received the data signed by the user with the correct LDP mechanism applied to it. This enables, for example, the protection of patient data used in a study to be guaranteed at the same time as the publication of the anonymized (signed) data to verify the statistical calculations made in the study. A natural extension of our work would be to instantiate our model with schemes allowing the use of other LDP mechanisms, such as the exponential mechanism. Another extension would be to find a way of adapting our primitive for use on non-discrete sets.

References

- [1] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.
- [2] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith, "What can we learn privately?" in *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 2008, pp. 531–540.
- [3] W. Liu, Y. Zhang, H. Yang, and Q. Meng, "A survey on differential privacy for medical data analysis," *Annals of Data Science*, pp. 1–15, 2023.
- [4] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Scientific Reports*, vol. 12, no. 1953, 2022.
- [5] J. Lee and C. Clifton, "How much is enough? choosing ϵ for differential privacy," in *Information Security, 14th International Conference, ISC 2011, Xi'an, China, October 26-29, 2011. Proceedings*, ser. Lecture Notes in Computer Science, X. Lai, J. Zhou, and H. Li, Eds., vol. 7001. Springer, 2011, pp. 325–340.
- [6] European Parliament and Council of Europe, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance)," *Official Journal*, no. L119, pp. 1–88, 2016.
- [7] A. Ambainis, M. Jakobsson, and H. Lipmaa, "Cryptographic randomized response techniques," in *Public Key Cryptography – PKC 2004*, F. Bao, R. Deng, and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 425–438.
- [8] A. Cheu, A. Smith, and J. Ullman, "Manipulation attacks in local differential privacy," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 883–900.
- [9] F. Kato, Y. Cao, and M. Yoshikawa, "Preventing manipulation attack in local differential privacy using verifiable randomization mechanism," in *Data and Applications Security and Privacy XXXV: 35th Annual IFIP WG 11.3 Conference, DBSec 2021, Calgary, Canada, July 19–20, 2021, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 43–60. [Online]. Available: https://doi.org/10.1007/978-3-030-81242-3_3
- [10] A. Biswas and G. Cormode, "Interactive proofs for differentially private counting," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, W. Meng, C. D. Jensen, C. Cremers, and E. Kirda, Eds. ACM, 2023, pp. 1919–1933. [Online]. Available: <https://doi.org/10.1145/3576915.3616681>
- [11] J. Böhrer and F. Kerschbaum, "Secure multi-party computation of differentially private median," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 2147–2164. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity20/presentation/boehler>
- [12] A. G. Sébert, M. Checri, O. Stan, R. Sirdey, and C. Gouy-Pailler, "Combining homomorphic encryption and differential privacy in federated learning," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, 2023, pp. 1–7.
- [13] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan, "Computational differential privacy," in *Advances in Cryptology - CRYPTO 2009*, S. Halevi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 126–142.
- [14] A. Roy Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha, "Crypte: Crypto-Assisted Differential Privacy on Untrusted Servers," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 603–619. [Online]. Available: <https://doi.org/10.1145/3318464.3380596>

- [15] G. Munilla Garrido, J. Sedlmeir, and M. Babel, “Towards verifiable differentially-private polling,” in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, ser. ARES '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://doi.org/10.1145/3538969.3538992>
- [16] M. Bellés-Muñoz, M. Isabel, J. L. Muñoz-Tapia, A. Rubio, and J. Baylina, “Circom: A circuit description language for building zero-knowledge applications,” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4733–4751, 2023.
- [17] J. Groth, “On the size of pairing-based non-interactive arguments,” in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 305–326.
- [18] C. Brzuska, M. Fischlin, T. Freudenreich, A. Lehmann, M. Page, J. Schelbert, D. Schröder, and F. Volk, “Security of sanitizable signatures revisited,” in *Public Key Cryptography–PKC 2009: 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18–20, 2009. Proceedings 12*. Springer, 2009, pp. 317–336.
- [19] X. Bultel and P. Lafourcade, “Unlinkable and strongly accountable sanitizable signatures from verifiable ring signatures,” in *Cryptology and Network Security: 16th International Conference, CANS 2017, Hong Kong, China, November 30–December 2, 2017, Revised Selected Papers*. Springer, 2018, pp. 203–226.
- [20] S. Canard and A. Jambert, “On extended sanitizable signature schemes,” in *Topics in Cryptology–CT-RSA 2010: The Cryptographers’ Track at the RSA Conference 2010, San Francisco, CA, USA, March 1–5, 2010. Proceedings*. Springer, 2010, pp. 179–194.
- [21] E. Boyle, S. Goldwasser, and I. Ivan, “Functional signatures and pseudorandom functions,” in *Public-Key Cryptography – PKC 2014*, H. Krawczyk, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 501–519.
- [22] M. Backes, S. Meiser, and D. Schröder, “Delegatable functional signatures,” in *Public-Key Cryptography – PKC 2016*, C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 357–386.
- [23] M. Bellare and G. Fuchsbauer, “Policy-based signatures,” in *Public-Key Cryptography – PKC 2014*, H. Krawczyk, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 520–537.
- [24] R. Johnson, D. A. Molnar, D. X. Song, and D. A. Wagner, “Homomorphic signature schemes,” in *The Cryptographer’s Track at RSA Conference*, 2002. [Online]. Available: <https://api.semanticscholar.org/CorpusID:13965939>
- [25] M. Blum, P. Feldman, and S. Micali, “Non-Interactive Zero-Knowledge and Its Applications,” in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 103–112. [Online]. Available: <https://doi.org/10.1145/62212.62222>
- [26] P. Kairouz, S. Oh, and P. Viswanath, “Extremal mechanisms for local differential privacy,” *J. Mach. Learn. Res.*, vol. 17, pp. 17:1–17:51, 2016.
- [27] V. Shoup, “Sequences of games: a tool for taming complexity in security proofs,” *Cryptology ePrint Archive*, Paper 2004/332, 2004, <https://eprint.iacr.org/2004/332>. [Online]. Available: <https://eprint.iacr.org/2004/332>
- [28] S. L. Warner, “Randomised response: a survey technique for eliminating evasive answer bias,” *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.
- [29] D. Chaum and T. P. Pedersen, “Wallet Databases with Observers,” in *Advances in Cryptology – CRYPTO’ 92*. Springer, 1993.
- [30] R. Cramer, I. Damgård, and B. Schoenmakers, “Proofs of partial knowledge and simplified design of witness hiding protocols,” in *Advances in Cryptology—CRYPTO’94: 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings*. Springer, 2001, pp. 174–187.
- [31] T. P. Pedersen, “Non-interactive and information-theoretic secure verifiable secret sharing,” in *Advances in Cryptology—CRYPTO’91: Proceedings*. Springer, 2001, pp. 129–140.
- [32] M. Fischlin and A. Mittelbach, “An overview of the hybrid argument,” *Cryptology ePrint Archive*, Paper 2021/088, 2021, <https://eprint.iacr.org/2021/088>. [Online]. Available: <https://eprint.iacr.org/2021/088>
- [33] X. Bultel and C. Olivier-Anclin, “Taming delegations in anonymous signatures: k-times anonymity for proxy and sanitizable signature,” in *Cryptology and Network Security*, M. Kohlweiss, R. Di Pietro, and A. Beresford, Eds. Singapore: Springer Nature Singapore, 2025, pp. 165–186.
- [34] A. Fiat and A. Shamir, “How To Prove Yourself: Practical Solutions to Identification and Signature Problems,” in *CRYPTO’ 86*. Springer, 1987.
- [35] J. Camenisch and V. Shoup, “Practical Verifiable Encryption and Decryption of Discrete Logarithms,” in *Advances in Cryptology - CRYPTO 2003*, D. Boneh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 126–144.
- [36] C. P. Schnorr, “Efficient identification and signatures for smart cards,” in *Advances in Cryptology — CRYPTO’ 89 Proceedings*, G. Brassard, Ed. New York, NY: Springer New York, 1990, pp. 239–252.
- [37] “Cryptographic Commitments on Anonymizable Data (Implementation).” [Online]. Available: <https://anonymous.4open.science/I/Cryptographic-Commitments-on-Anonymizable-Data-D8E8>
- [38] I. A. Lovecruft and H. de Valence, “curve25519_dalek.” [Online]. Available: https://docs.rs/curve25519-dalek/latest/curve25519_dalek/

Appendix A. Proof of Theorem 1

Proof. By abuse of notation, we simply note $\Pr[b = b_*]$ the probability:

$$\Pr \left[\begin{array}{l} b \stackrel{\$}{\leftarrow} \{0, 1\}; \hat{m} \leftarrow \text{LDP}(m_b); \\ b_* \leftarrow \text{OptiGuess}(\lambda, \text{LDP}, \hat{m}, m_0, m_1) \end{array} : b = b_* \right].$$

When b is set to a value $x \in \{0, 1\}$ (resp. \hat{m} to a value $\hat{x} \in \mathcal{M}$), we will note $\Pr[b = b_* | b = x]$ (resp. $\Pr[b = b_* | \hat{m} = \hat{x}]$). Since $\Pr[b = 0] = \Pr[b = 1] = \frac{1}{2}$, we have:

$$\begin{aligned} \Pr[b = b_*] &= \Pr[b = 0] \cdot \Pr[b = b_* | b = 0] + \\ &\quad \Pr[b = 1] \cdot \Pr[b = b_* | b = 1] \\ &= \frac{1}{2} (\Pr[b = b_* | b = 0] + \Pr[b = b_* | b = 1]). \end{aligned}$$

We first evaluate the first probability:

$$\begin{aligned} \Pr[b = b_* | b = 0] &= \sum_{\hat{x} \in \mathcal{M}} \Pr[\hat{m} = \hat{x} | b = 0] \cdot \Pr[b = b_* | b = 0 \wedge \hat{m} = \hat{x}] \\ &= \sum_{\hat{x} \in \mathcal{M}} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] \cdot \Pr[b = b_* | b = 0 \wedge \hat{m} = \hat{x}] \end{aligned}$$

For all comparison operators $\square \in \{=, <, >\}$, we define the set \mathcal{M}_\square as follows:

$$\hat{x} \in \mathcal{M}_\square \Leftrightarrow \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] \square \Pr[\hat{x} \leftarrow \text{LDP}(m_1)].$$

Let $\mathcal{P} = \{\mathcal{M}_=, \mathcal{M}_<, \mathcal{M}_>\}$ be a set, we have that \mathcal{P} is a partition of \mathcal{M} . We recall that by definition, OptiGuess returns 0 with probability:

- 1 if $\Pr[\hat{x} \leftarrow \text{LDP}(m_0)] > \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]$,
- $\frac{1}{2}$ if $\Pr[\hat{x} \leftarrow \text{LDP}(m_0)] = \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]$, and
- 0 if $\Pr[\hat{x} \leftarrow \text{LDP}(m_0)] < \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]$.

We deduce that:

$$\begin{aligned}
& \Pr[b = b_* | b = 0] \\
&= \sum_{\hat{x} \in \mathcal{M}} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] \cdot \Pr[b = b_* | b = 0 \wedge \hat{m} = \hat{x}] \\
&= \sum_{S \in \mathcal{P}} \sum_{\hat{x} \in S} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] \cdot \Pr \left[b = b_* \mid \begin{array}{l} b = 0 \wedge \\ \hat{m} = \hat{x} \end{array} \right] \\
&= \sum_{\hat{x} \in \mathcal{M}_>} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] + \\
&\quad \frac{1}{2} \cdot \sum_{\hat{x} \in \mathcal{M}_=} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] + \\
&\quad 0 \cdot \sum_{\hat{x} \in \mathcal{M}_<} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] \\
&= \sum_{\hat{x} \in \mathcal{M}_>} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] + \\
&\quad \frac{1}{2} \sum_{\hat{x} \in \mathcal{M}_=} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)].
\end{aligned}$$

In a similar way, we have:

$$\begin{aligned}
\Pr[b = b_* | b = 1] &= \sum_{\hat{x} \in \mathcal{M}_<} \Pr[\hat{x} \leftarrow \text{LDP}(m_1)] + \\
&\quad \frac{1}{2} \sum_{\hat{x} \in \mathcal{M}_=} \Pr[\hat{x} \leftarrow \text{LDP}(m_1)].
\end{aligned}$$

Furthermore:

- for all $\hat{x} \in \mathcal{M}_>$,
$$\max(\Pr[\hat{x} \leftarrow \text{LDP}(m_0)], \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]) = \Pr[\hat{x} \leftarrow \text{LDP}(m_0)].$$
- for all $\hat{x} \in \mathcal{M}_=$,
$$\max(\Pr[\hat{x} \leftarrow \text{LDP}(m_0)], \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]) = \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] = \Pr[\hat{x} \leftarrow \text{LDP}(m_1)].$$
- for all $\hat{x} \in \mathcal{M}_<$,
$$\max(\Pr[\hat{x} \leftarrow \text{LDP}(m_0)], \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]) = \Pr[\hat{x} \leftarrow \text{LDP}(m_1)].$$

We deduce:

$$\begin{aligned}
& \Pr[b = b_* | b = 0] + \Pr[b = b_* | b = 1] \\
&= \sum_{\hat{x} \in \mathcal{M}_>} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] + \\
&\quad \frac{1}{2} \sum_{\hat{x} \in \mathcal{M}_=} \Pr[\hat{x} \leftarrow \text{LDP}(m_0)] + \\
&\quad \sum_{\hat{x} \in \mathcal{M}_<} \Pr[\hat{x} \leftarrow \text{LDP}(m_1)] + \\
&\quad \frac{1}{2} \sum_{\hat{x} \in \mathcal{M}_=} \Pr[\hat{x} \leftarrow \text{LDP}(m_1)] \\
&= \sum_{\hat{x} \in \mathcal{M}} \max(\Pr[\hat{x} \leftarrow \text{LDP}(m_0)], \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]).
\end{aligned}$$

This leads to the result of the theorem:

$$\begin{aligned}
& \Pr[b = b_*] \\
&= \frac{1}{2} \sum_{\hat{x} \in \mathcal{M}} \max(\Pr[\hat{x} \leftarrow \text{LDP}(m_0)], \Pr[\hat{x} \leftarrow \text{LDP}(m_1)]).
\end{aligned}$$

This concludes the proof. \square

Appendix B. Useful Lemmas

We first give a technical lemma, which will be useful to simplify the winning probabilities for the adversary.

Lemma 1. *Let $\text{Exp}_{\mathcal{A},b}(\lambda)$ be an experiment depending on a bit b and a security parameter λ for an adversary $\mathcal{A} \in \text{POLY}(\lambda)$ that returns a bit. we have:*

$$\begin{aligned}
& \left| \Pr \left[b \stackrel{\$}{\leftarrow} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A},b}(\lambda) \right] - \frac{1}{2} \right| \\
&= \frac{1}{2} \left| (\Pr[1 \leftarrow \text{Exp}_{\mathcal{A},1}(\lambda)] - \Pr[0 \leftarrow \text{Exp}_{\mathcal{A},0}(\lambda)]) \right|.
\end{aligned}$$

Proof.

$$\begin{aligned}
& \left| \Pr \left[b \stackrel{\$}{\leftarrow} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A},b}(\lambda) \right] - \frac{1}{2} \right| \\
&= \left| \Pr \left[b \stackrel{\$}{\leftarrow} \{0, 1\} : b = 0 \right] \cdot \Pr[1 \leftarrow \text{Exp}_{\mathcal{A},0}(\lambda)] + \right. \\
&\quad \left. \Pr \left[b \stackrel{\$}{\leftarrow} \{0, 1\} : b = 1 \right] \cdot \Pr[1 \leftarrow \text{Exp}_{\mathcal{A},1}(\lambda)] - \frac{1}{2} \right| \\
&= \left| \frac{1}{2} (1 - \Pr[0 \leftarrow \text{Exp}_{\mathcal{A},0}(\lambda)]) + \right. \\
&\quad \left. \Pr[1 \leftarrow \text{Exp}_{\mathcal{A},1}(\lambda)] - \frac{1}{2} \right| \\
&= \frac{1}{2} \left| (\Pr[1 \leftarrow \text{Exp}_{\mathcal{A},1}(\lambda)] - \Pr[0 \leftarrow \text{Exp}_{\mathcal{A},0}(\lambda)]) \right|.
\end{aligned}$$

\square

We now give some results needed in the security analysis of our efficient LDP commitment scheme (Section 4.4).

Lemma 2. *Let $\mathbb{G} = \langle g \rangle$ be multiplicative group of prime order p . Let \mathcal{A} be a PPT algorithm, consider the LOR-DDH experiment $\text{Exp}_{\mathcal{A},b}^{\text{LOR-DDH}}(\lambda)$ as follows:*

$$\begin{aligned}
& \text{Exp}_{\mathcal{A},b}^{\text{LOR-DDH}}(\lambda): \\
& (x, y_0, y_1) \stackrel{\$}{\leftarrow} (\mathbb{Z}_p^*)^3 \\
& z \leftarrow x \cdot y_b \\
& b' \leftarrow \mathcal{A}(g^x, g^{y_0}, g^{y_1}, g^z) \\
& \text{return } b = b'
\end{aligned}$$

Under the DDH assumption, there exists a negligible function $\epsilon_{\text{LOR-DDH}}$ s.t. for any PPT algorithm \mathcal{A} :

$$\begin{aligned}
& \left| \Pr[0 \leftarrow \text{Exp}_{\mathcal{A},0}^{\text{LOR-DDH}}(\lambda)] - \Pr[1 \leftarrow \text{Exp}_{\mathcal{A},1}^{\text{LOR-DDH}}(\lambda)] \right| \\
&\leq \epsilon_{\text{LOR-DDH}}(\lambda).
\end{aligned}$$

Proof. Game G_0 : This game is the same as the LOR-DDH experiment in the case where $b \stackrel{\$}{\leftarrow} \{0, 1\}$.

$$\begin{aligned}
& \Pr[\mathcal{A} \text{ wins the game } G_0] \\
&= \Pr \left[b \stackrel{\$}{\leftarrow} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A},b}^{\text{LOR-DDH}}(\lambda) \right].
\end{aligned}$$

Game G_1 : This game is the same as the game G_0 except that the challenger replaces g^z by a random element.

Let $\epsilon_{\text{DDH}}(\lambda)$ be the maximum DDH advantage of all PPT algorithm \mathcal{D} , we claim that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq 2 \cdot \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

We prove this claim by reduction.

Assume there exists a PPT algorithm \mathcal{A} such that $\epsilon_{\text{LOR-DDH}}$ is non negligible, we build a PPT distinguishing algorithm \mathcal{D} against the DDH assumption as follows:

Algorithm $\mathcal{D}(g^x, g^y, g^\gamma)$: Parses g^x as X , g^y as Y_0 and g^γ as Γ , picks $r \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $Y_1 \leftarrow g^r$. \mathcal{D} picks $b_* \xleftarrow{\$} \{0, 1\}$, and runs $b' \leftarrow \mathcal{A}(X, Y_{b_*}, Y_{1-b_*}, \Gamma)$.

- If $b_* = b'$, then returns 1.
- Else, then returns 0.

We have:

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_0] \\ & = \Pr[(x, y) \xleftarrow{\$} (\mathbb{Z}_p^*)^2 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^{x \cdot y})], \end{aligned}$$

and

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_1] \\ & = \Pr[(x, y, z) \xleftarrow{\$} (\mathbb{Z}_p^*)^3 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^z)]. \end{aligned}$$

Thus, we deduce that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

We claim that:

$$\Pr[\mathcal{A} \text{ wins the game } G_1] = \frac{1}{2}.$$

We note that in the game G_0 , the algorithm \mathcal{A} receives the tuple $(g^x, g^{y_0}, g^{y_1}, g^{x \cdot y_b})$ and it has to guess the bit b randomly picked by the challenger. The proof of the claim derives from the fact that in the game G_1 , the algorithm \mathcal{A} receives the tuple $(g^x, g^{y_0}, g^{y_1}, g^z)$ where g^z is a random group element, so \mathcal{A} can not deduce any information on the bit b and it returns a random bit $b' \in \{0, 1\}$, we have:

$$\Pr[\mathcal{A} \text{ wins the game } G_1] = \frac{1}{2}.$$

We have:

$$\begin{aligned} & \left| \Pr \left[b \xleftarrow{\$} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}}(\lambda) \right] - \frac{1}{2} \right| \\ & = |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

From Lemma 1, we obtain:

$$\begin{aligned} & \left| \Pr \left[b \xleftarrow{\$} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}}(\lambda) \right] - \frac{1}{2} \right| \\ & = \frac{1}{2} \left| \Pr \left[1 \leftarrow \text{Exp}_{\mathcal{A}, 1}^{\text{LOR-DDH}}(\lambda) \right] - \Pr \left[0 \leftarrow \text{Exp}_{\mathcal{A}, 0}^{\text{LOR-DDH}}(\lambda) \right] \right|. \end{aligned}$$

Finally, we have:

$$\begin{aligned} & \left| \Pr \left[1 \leftarrow \text{Exp}_{\mathcal{A}, 1}^{\text{LOR-DDH}}(\lambda) \right] - \Pr \left[0 \leftarrow \text{Exp}_{\mathcal{A}, 0}^{\text{LOR-DDH}}(\lambda) \right] \right| \\ & = 2 \cdot \left| \Pr \left[b \xleftarrow{\$} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}}(\lambda) \right] - \frac{1}{2} \right| \\ & \leq 2 \cdot \epsilon_{\text{DDH}}(\lambda), \end{aligned}$$

which concludes the proof of Lemma 3. \square

Lemma 3. Let $\mathbb{G} = \langle g \rangle$ be multiplicative group of prime order p . Let \mathcal{A} be a PPT algorithm, consider the LOR-DDH - 2 experiment $\text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}-2}(\lambda)$ as follows:

$$\begin{aligned} & \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}-2}(\lambda): \\ & (x, y_0, y_1) \xleftarrow{\$} (\mathbb{Z}_p^*)^3 \\ & z_0 \leftarrow x \cdot y_b \\ & z_1 \leftarrow x \cdot y_{1-b} \\ & b' \leftarrow \mathcal{A}(g^x, g^{y_0}, g^{y_1}, g^{z_0}, g^{z_1}) \\ & \text{return } b = b' \end{aligned}$$

Under the DDH assumption, there exists a negligible function $\epsilon_{\text{LOR-DDH}-2}$ s.t. for any PPT algorithm \mathcal{A} :

$$\begin{aligned} & \left| \Pr \left[0 \leftarrow \text{Exp}_{\mathcal{A}, 0}^{\text{LOR-DDH}-2}(\lambda) \right] - \Pr \left[1 \leftarrow \text{Exp}_{\mathcal{A}, 1}^{\text{LOR-DDH}-2}(\lambda) \right] \right| \leq \epsilon_{\text{LOR-DDH}-2}(\lambda). \end{aligned}$$

Proof. **Game** G_0 : This game is the same as the LOR-DDH - 2 experiment in the case where $b \xleftarrow{\$} \{0, 1\}$.

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_0] \\ & = \Pr \left[b \xleftarrow{\$} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}-2}(\lambda) \right]. \end{aligned}$$

Game G_1 : This game is the same as the game G_0 except that the challenger replaces g^{z_0} by a random element. We claim that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

We prove this claim by reduction.

Assume there exists a PPT algorithm \mathcal{A} such that $\epsilon_{\text{LOR-DDH}-2}$ is non negligible, we build a PPT distinguishing algorithm \mathcal{D} against the DDH assumption as follows:

Algorithm $\mathcal{D}(g^x, g^y, g^\gamma)$: picks $r \xleftarrow{\$} \mathbb{Z}_p^*$, Parses g^x as X , g^y as Y_0 and g^γ as Z_0 , and sets $Y_1 \leftarrow g^r$ and $Z_1 \leftarrow X^r$. \mathcal{D} picks $b_* \xleftarrow{\$} \{0, 1\}$ and runs $b' \leftarrow \mathcal{A}(X, Y_{b_*}, Y_{1-b_*}, Z_0, Z_1)$.

- If $b_* = b'$, returns 1.
- Else, returns 0.

We have:

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_0] \\ & = \Pr[(x, y) \xleftarrow{\$} (\mathbb{Z}_p^*)^2 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^{x \cdot y})], \end{aligned}$$

and

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_1] \\ & = \Pr[(x, y, z) \xleftarrow{\$} (\mathbb{Z}_p^*)^3 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^z)]. \end{aligned}$$

Thus, we deduce that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq \epsilon_{\text{DDH}}(\lambda), \end{aligned}$$

Game G_2 : This game is the same as the game G_1 except that the challenger replaces g^{z_1} by a random element. Let $\epsilon_{\text{DDH}}(\lambda)$ be the maximum DDH advantage of all PPT algorithm \mathcal{D} , we claim that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ & \leq \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

We prove this claim by reduction.

Assume there exists a PPT algorithm \mathcal{A} such that $\epsilon_{\text{LOR-DDH-2}}$ is non negligible, we build a PPT distinguishing algorithm \mathcal{D} against the DDH assumption as follows:

Algorithm $\mathcal{D}(g^x, g^y, g^z)$: picks $r, w \xleftarrow{\$} \mathbb{Z}_p^{*2}$, parses g^x as X , g^y as Y_0 and g^z as Z_1 and sets $Y_1 \leftarrow g^r$ and $Z_0 \leftarrow X^w$. \mathcal{D} picks $b_* \xleftarrow{\$} \{0, 1\}$ and runs $b' \leftarrow \mathcal{A}(X, Y_{b_*}, Y_{1-b_*}, Z_0, Z_1)$.

- If $b_* = b'$, returns 1.
- Else, returns 0.

We have:

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_1] \\ & = \Pr[(x, y) \xleftarrow{\$} (\mathbb{Z}_p^*)^2 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^{x \cdot y})], \end{aligned}$$

and

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_2] \\ & = \Pr[(x, y, z) \xleftarrow{\$} (\mathbb{Z}_p^*)^3 : 1 \leftarrow \mathcal{D}(g^x, g^y, g^z)]. \end{aligned}$$

Thus, we deduce that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ & \leq \epsilon_{\text{DDH}}(\lambda), \end{aligned}$$

We claim that:

$$\Pr[\mathcal{A} \text{ wins the game } G_2] = \frac{1}{2}.$$

We note that in the game G_0 , the algorithm \mathcal{A} receives the tuple $(g^x, g^{y_0}, g^{y_1}, g^{x \cdot y_b}, g^{x \cdot y_{1-b}})$ and it has to guess the bit b randomly picked by the challenger. The proof of the claim derives from the fact that in the game G_2 , the algorithm \mathcal{A} receives the tuple $(g^x, g^{y_0}, g^{y_1}, g^{z_0}, g^{z_1})$ where g^{z_0} and g^{z_1} are random group elements, so \mathcal{A} can not deduce any information on the bit b and it returns a random bit $b' \in \{0, 1\}$, we have:

$$\Pr[\mathcal{A} \text{ wins the game } G_2] = \frac{1}{2}.$$

We obtain that:

$$\begin{aligned} & \left| \Pr \left[b \xleftarrow{\$} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH}}(\lambda) \right] - \frac{1}{2} \right| \\ & = |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ & \leq 2 \cdot \epsilon_{\text{DDH}}(\lambda). \end{aligned}$$

Finally, we have:

$$\begin{aligned} & \left| \Pr \left[1 \leftarrow \text{Exp}_{\mathcal{A}, 1}^{\text{LOR-DDH-2}}(\lambda) \right] - \Pr \left[0 \leftarrow \text{Exp}_{\mathcal{A}, 0}^{\text{LOR-DDH-2}}(\lambda) \right] \right| \\ & = 2 \cdot \left| \Pr \left[b \xleftarrow{\$} \{0, 1\} : 1 \leftarrow \text{Exp}_{\mathcal{A}, b}^{\text{LOR-DDH-2}}(\lambda) \right] - \frac{1}{2} \right| \\ & \leq 4 \cdot \epsilon_{\text{DDH}}(\lambda), \end{aligned}$$

which concludes the proof of the lemma. \square

We recall the Discrete Logarithm (DL) assumptions in a group $\mathbb{G} = \langle g \rangle$ of prime order p . The DL assumption states that given a random $x \in \mathbb{Z}_p^*$, the probability that any PPT algorithm guesses x on input g^x is negligible.

Definition 14 (Discrete Logarithm (DL) assumption). *Let $\mathbb{G} = \langle g \rangle$ be a multiplicative group of prime order p . The Discrete Logarithm (DL) assumption states that there exists a negligible function ϵ_{DL} such that for any PPT algorithm \mathcal{A} , we have $\Pr \left[h \xleftarrow{\$} \mathbb{G}; x \leftarrow \mathcal{A}(h) : g^x = h \right] \leq \epsilon_{\text{DL}}(\lambda)$.*

Lemma 4. *Let λ be a security parameter, \mathbb{G} be a group of prime order p , and $g \in \mathbb{G}$ be a generator. If the DL assumption holds, there exists a negligible function $\epsilon_{\ell\text{-col}}$ such that for all PPT algorithm $\mathcal{A} \in \text{POLY}(\lambda)$ and all integer ℓ :*

$$\begin{aligned} & \Pr \left[\begin{array}{l} (g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \xleftarrow{\$} (\mathbb{G}^\ell)^2; \\ (x_1, x_2) \leftarrow \mathcal{A} \left((g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \right); \end{array} : \begin{array}{l} \left(\prod_{i=1}^{\ell} g_{(i,x_1[i])} \right) \\ = \prod_{i=1}^{\ell} g_{(i,x_2[i])} \\ \wedge (x_1 \neq x_2) \end{array} \right] \\ & \leq \epsilon_{\ell\text{-col}}(\lambda). \end{aligned}$$

Proof. We define the games G_0 and G_1 as follows:

Game G_0 :

$$\begin{aligned} & (g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \xleftarrow{\$} (\mathbb{G}^\ell)^2; \\ & (x_1, x_2) \leftarrow \mathcal{A} \left((g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \right); \end{aligned}$$

If $\left(\prod_{i=1}^{\ell} g_{(i,x_1[i])} = \prod_{i=1}^{\ell} g_{(i,x_2[i])} \right) \wedge (x_1 \neq x_2)$, then return 1, else return 0.

Game G_1 :

$$\begin{aligned} & k \xleftarrow{\$} [\ell]; (g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \xleftarrow{\$} \mathbb{G}^{2\ell}; \\ & (x_1, x_2) \leftarrow \mathcal{A} \left((g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \right); \end{aligned}$$

If $x_1[k] = x_2[k]$ then abort the game and return 0;

If $\left(\prod_{i=1}^{\ell} g_{(i,x_1[i])} = \prod_{i=1}^{\ell} g_{(i,x_2[i])} \right)$, then return 1, else return 0.

We claim that, for all PPT algorithm $\mathcal{A} \in \text{POLY}(k)$:

- 1) $\Pr[\mathcal{A} \text{ wins } G_0] \leq \ell \cdot \Pr[\mathcal{A} \text{ wins } G_1]$
- 2) $\Pr[\mathcal{A} \text{ wins } G_1] \leq \epsilon_{\text{DL}}(\lambda)$

Proof of Claim 1: G_1 is defined as G_0 except that the challenger picks k at random in $[\ell]$ and aborts if $x_1[k] = x_2[k]$. First remark that if there is no index k' such that $x_1[k'] \neq x_2[k']$, then \mathcal{A} does not win the game G_0 because $x_1 = x_2$. In other words, if \mathcal{A} wants to have a chance of winning, they chooses x_1 and x_2

in such a way that there is at least an index k' such that $x_1[k'] \neq x_2[k']$. The game G_1 does not abort if the challenger guesses this index by picking k . Finally, The adversary increases its winning advantage by a factor equalling the probability of guessing correctly k , *i.e.*, at least $1/\ell$.

Proof of Claim 2: We prove this claim by reduction. We build an algorithm $\mathcal{B} \in \text{POLY}(\lambda)$ that tries to return the discrete logarithm of a random group element by using \mathcal{A} as a black box.

Algorithm $\mathcal{B}(h)$: Pick $(\alpha_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \xleftarrow{\$} (\mathbb{Z}_p^*)^{2\ell}$ and $k \xleftarrow{\$} [\ell]$, then set $g_{(k,0)} \leftarrow h$. For all (i,j) in $([\ell] \times \{0,1\}) \setminus \{(h,0)\}$, set $g_{i,j} \leftarrow g^{\alpha_{(i,j)}}$. Run $(x_1, x_2) \leftarrow \mathcal{A} \left((g_{(i,j)})_{\substack{i \in [\ell] \\ j \in \{0,1\}}} \right)$. If $x_1[k] = x_2[k]$, then abort and return 0. If $x_1[k] = 0$ then set $(x, \bar{x}) \leftarrow (x_1, x_2)$, else set $(x, \bar{x}) \leftarrow (x_2, x_1)$. We remark that $x[k] = 0$ and $\bar{x}[k] = 1$, so $\alpha_{(k,\bar{x}[k])}$ is defined but $\alpha_{(k,x[k])}$ is not defined. Set then return:

$$x_* \leftarrow \left(\sum_{i=1}^{\ell} \alpha_{(i,\bar{x}[i])} - \sum_{\substack{i=1 \\ i \neq k}}^{\ell} \alpha_{(i,x[i])} \right)$$

We first remark that if h is chosen at random in the uniform distribution on \mathbb{G} , then the game G_1 is perfectly simulated for \mathcal{A} . If \mathcal{A} wins the simulated game G_1 , it holds that $(\prod_{i=1}^{\ell} g_{(i,x_1[i])} = \prod_{i=1}^{\ell} g_{(i,x_2[i])})$ and $x_1[k] \neq x_2[k]$, which implies that $(\prod_{i=1}^{\ell} g_{(i,\bar{x}[i])} = g_{(k,x[k])} \cdot \prod_{\substack{i=1 \\ i \neq k}}^{\ell} g_{(i,x[i])})$. Moreover, we have $x[k] = 0$, so $g_{(k,x[k])} = g_{(k,0)} = h$. We deduce that:

$$\begin{aligned} g^{x_*} &= g \left(\sum_{i=1}^{\ell} \alpha_{(i,\bar{x}[i])} - \sum_{\substack{i=1 \\ i \neq k}}^{\ell} \alpha_{(i,x[i])} \right) \\ &= \frac{g^{\sum_{i=1}^{\ell} \alpha_{(i,\bar{x}[i])}}}{g^{\sum_{\substack{i=1 \\ i \neq k}}^{\ell} \alpha_{(i,x[i])}}} = \frac{\prod_{i=1}^{\ell} g_{(i,\bar{x}[i])}}{\prod_{\substack{i=1 \\ i \neq k}}^{\ell} g_{(i,x[i])}} = h \end{aligned}$$

Conclusion: From the two claims, we deduce $\Pr[\mathcal{A} \text{ wins } G_0] \leq \ell \cdot \epsilon_{\text{DL}}(\lambda)$, which conclude the proof since $\ell \cdot \epsilon_{\text{DL}}(\lambda)$ is negligible. \square

Appendix C.

Proof of Theorem 2

We show that the if a LDP-C scheme P is Hiding and ROS-LDP-Hiding, then P is IND-LDP-Hiding.

Proof. We define the games G_0 , G_1 , G_2 and G_3 as follows:

Game G_0 :
 $\text{set} \leftarrow \text{Setup}(\lambda, \text{LDP})$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$
 $b' \xleftarrow{\$} \{0, 1\}$
 $\theta \xleftarrow{\$} \Theta$
 $(k, c, \pi_*) \leftarrow \text{Commit}(m_{b'}, \theta)$

$\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$
 $(\hat{m}, \hat{\pi}) \leftarrow \text{OpenLDP}(k, c, \hat{\theta})$
 $b_* \leftarrow \mathcal{A}_3(\hat{m}, \hat{\pi})$
 return $b_* = b'$

Game G_1 :

$\text{set} \leftarrow \text{Setup}(\lambda, \text{LDP})$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$
 $b' \xleftarrow{\$} \{0, 1\}$
 $\theta \xleftarrow{\$} \Theta$
 $(k, c, \pi_*) \leftarrow \text{Commit}(m_{b'}, \theta)$
 $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$
 $(\hat{m}_0, \hat{\pi}_0) \leftarrow \text{OpenLDP}(k, c, \hat{\theta})$
 $\hat{\pi}_1 \leftarrow \text{Sim}(\hat{m}_0)$
 $b_* \leftarrow \mathcal{A}_3(\hat{m}_0, \hat{\pi}_1)$ return $b_* = b'$

Game G_2 :

$\text{set} \leftarrow \text{Setup}(\lambda, \text{LDP})$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$
 $b' \xleftarrow{\$} \{0, 1\}$
 $\theta \xleftarrow{\$} \Theta$
 $(k, c, \pi_*) \leftarrow \text{Commit}(m_{b'}, \theta)$
 $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$
 $\hat{m}_1 \leftarrow \text{LDP}(m_{b'})$
 $\hat{\pi}_1 \leftarrow \text{Sim}(\hat{m}_1)$
 $b_* \leftarrow \mathcal{A}_3(\hat{m}_1, \hat{\pi}_1)$
 return $b_* = b'$

Game G_3 :

$\text{set} \leftarrow \text{Setup}(\lambda, \text{LDP})$
 $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$
 $b' \xleftarrow{\$} \{0, 1\}$
 $\theta \xleftarrow{\$} \Theta$
 $m \xleftarrow{\$} \mathcal{M}$
 $(k, c, \pi_*) \leftarrow \text{Commit}(m, \theta)$
 $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$
 $\hat{m}_1 \leftarrow \text{LDP}(m_{b'})$
 $\hat{\pi}_1 \leftarrow \text{Sim}(\hat{m}_1)$
 $b_* \leftarrow \mathcal{A}_3(\hat{m}_1, \hat{\pi}_1)$
 return $b_* = b'$

We claim that, for all PPT algorithm $\mathcal{A} \in \text{POLY}(k)$:

- 1) $\Pr[\mathcal{A} \text{ wins the game } G_0] = \Pr[\mathcal{A} \text{ wins the game } G_1]$.
- 2) $|\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \leq \epsilon_{\text{ROS-LDP-Hiding}}(\lambda)$.
- 3) $|\Pr[\mathcal{A} \text{ wins the game } G_2] - \Pr[\mathcal{A} \text{ wins the game } G_3]| \leq \frac{1}{2} \cdot \epsilon_{\text{Hiding}}(\lambda)$.

Proof of Claim C: Since the proof $\hat{\pi}$ is zero-knowledge, we have:

$$\Pr[\mathcal{A} \text{ wins the game } G_0] = \Pr[\mathcal{A} \text{ wins the game } G_1].$$

Proof of Claim C: We prove this claim by reduction.

$\mathcal{B}_1(\text{set})$:
 $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$
 $b' \leftarrow \{0, 1\}$
 return $m_{b'}$

$\mathcal{B}_2(c, \pi_*)$:
 $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$

return $\hat{\theta}$

Let b be the bit that \mathcal{B} has to guess.

$\mathcal{B}_3(\hat{m}_b, \hat{\pi}_b)$:
 $b_* \leftarrow \mathcal{A}_3(\hat{m}_b, \hat{\pi}_b)$.
 if $b_* = b'$, return $b_{**} = 1$
 else, return $b_{**} = 0$

We have:

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_1] \\ &= \Pr[b_{**} = 1 | b = 0] \\ &= \Pr[\mathcal{B} \text{ returns 1 in } \text{Exp}_{P, \text{LDP}, \mathcal{B}, 0}^{\text{ROS-LDP-Hiding}}(\lambda)] \\ &= \Pr[0 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 0}^{\text{ROS-LDP-Hiding}}(\lambda)], \end{aligned}$$

and

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_2] \\ &= \Pr[b_{**} = 1 | b = 1] \\ &= \Pr[\mathcal{B} \text{ returns 1 in } \text{Exp}_{P, \text{LDP}, \mathcal{B}, 1}^{\text{ROS-LDP-Hiding}}(\lambda)] \\ &= \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 1}^{\text{ROS-LDP-Hiding}}(\lambda)]. \end{aligned}$$

Thus, we have:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ &= \left| \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 0}^{\text{ROS-LDP-Hiding}}(\lambda)] - \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 1}^{\text{ROS-LDP-Hiding}}(\lambda)] \right| \\ &\leq \epsilon_{\text{ROS-LDP-Hiding}}(\lambda). \end{aligned}$$

Proof of Claim C: We prove this claim by reduction.

$\mathcal{B}_1(\text{set})$:
 $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$
 $b' \xleftarrow{\$} \{0, 1\}$
 $\tilde{m}_0 \leftarrow m_{b'}$
 $\tilde{m}_1 \xleftarrow{\$} \mathcal{M}$
 return $(\tilde{m}_0, \tilde{m}_1)$

Let b be the bit that \mathcal{B} has to guess.

$\mathcal{B}_2(c, \pi_*)$:
 $\hat{\theta} \leftarrow \mathcal{A}_2(c, \pi_*)$
 $\hat{m} \leftarrow \text{LDP}(\tilde{m}_b)$
 $\hat{\pi} \leftarrow \text{Sim}(\hat{m})$
 $b_* \leftarrow \mathcal{A}_3(\hat{m}, \hat{\pi})$
 if $b_* = b'$ return $b_{**} = 1$
 else, return $b_{**} = 0$

We have:

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_2] \\ &= \Pr[b_{**} = 1 | b = 0] \\ &= \Pr[\mathcal{B} \text{ returns 1 in } \text{Exp}_{P, \text{LDP}, \mathcal{B}, 0}^{\text{Hiding}}(\lambda)] \\ &= \Pr[0 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 0}^{\text{Hiding}}(\lambda)], \end{aligned}$$

and

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_3] \\ &= \Pr[b_{**} = 1 | b = 1] \\ &= \Pr[\mathcal{B} \text{ returns 1 in } \text{Exp}_{P, \text{LDP}, \mathcal{B}, 1}^{\text{Hiding}}(\lambda)] \\ &= \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 1}^{\text{Hiding}}(\lambda)]. \end{aligned}$$

Thus, we have:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_2] - \Pr[\mathcal{A} \text{ wins the game } G_3]| \\ &= \left| \Pr[0 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 0}^{\text{Hiding}}(\lambda)] - \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{B}, 1}^{\text{Hiding}}(\lambda)] \right| \\ &\leq \epsilon_{\text{Hiding}}(\lambda). \end{aligned}$$

We remark that the game G_0 is the same as the IND-LDP-Hiding experiment in the case where $b = 0$, we have:

$$\begin{aligned} & \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{IND-LDP-Hiding}}(\lambda)] \\ &= \Pr[\mathcal{A} \text{ wins the game } G_0]. \end{aligned}$$

In the game G_3 , the algorithm \mathcal{A} can not deduce any information on the bit b' knowing the commitment c which is the commitment of a random message, so the optimal strategy for \mathcal{A} to win is to use the algorithm OptiGuess, then we have:

$$\begin{aligned} & \Pr[\mathcal{A} \text{ wins the game } G_3] \\ &\leq \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{IND-LDP-Hiding}}(\lambda)]. \end{aligned}$$

Finally, we get:

$$\begin{aligned} & \left| \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 0}^{\text{IND-LDP-Hiding}}(\lambda)] - \Pr[1 \leftarrow \text{Exp}_{P, \text{LDP}, \mathcal{A}, 1}^{\text{IND-LDP-Hiding}}(\lambda)] \right| \\ &= |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_3]| \\ &\leq \epsilon_{\text{ROS-LDP-Hiding}}(\lambda) + \epsilon_{\text{Hiding}}(\lambda). \end{aligned}$$

□

Appendix D.

Proof of Theorem 3 (Security Proofs for ORRC)

The proof of Theorem 3 for the scheme ORRC follows from the following lemmas.

Lemma 5. *The ORRC scheme instantiated with zero-knowledge proofs and a group where the DDH assumption holds is hiding.*

Lemma 6. *The ORRC scheme instantiated with zero-knowledge proofs and a group where the DDH assumption holds is ROS-LDP-hiding.*

Lemmas 5 and 6 implies that the ORRC scheme is IND-LDP-hiding.

Lemma 7. *The ORRC scheme instantiated with extractable proofs and a group where the DDH assumption holds is binding.*

Lemma 8. *The ORRC scheme instantiated with extractable proofs and a group where the DDH assumption holds is LDP-binding.*

Lemma 9. *The ORRC scheme instantiated with extractable proofs and a group where the DDH assumption holds is probabilistic-LDP-binding.*

D.1. Proof of Lemma 5 for ORRC

An adversary \mathcal{A} wins the Hiding experiment by distinguishing between two messages which one has been committed. In other words, since our scheme, we use an encoding based on the discrete logarithm assumption for the commitment, that means the adversary wins the experiment by deducing the original message from the encoding. To prove the lemma, we use the following sequence of games [27] based on indistinguishability, where the first game is the Hiding experiment given in the case where $b = 0$ and the last game is the Hiding experiment in the case where $b = 1$.

Proof. **Game G_0 :** This game is the same as the Hiding experiment in the case where $b = 0$, we have:

$$\Pr[\mathcal{A} \text{ returns 1 in } G_0] = \Pr\left[0 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},0}^{\text{Hiding}}(\lambda)\right].$$

Game G_1 : This game is the same as the game G_0 except that the challenger replaces the zero-knowledge proof π_* by a simulated proof, we have:

$$\Pr[\mathcal{A} \text{ returns 1 in } G_0] = \Pr[\mathcal{A} \text{ returns 1 in } G_1].$$

In the game G_1 , the challenger commits the message m_0 , we parse the input of the adversary c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$. We note that the information of the i -th bit of the message m_0 is contained in $A_{(i,2)}$.

Game G_2 : This game is the same as the Hiding experiment in the case where $b = 1$, we have:

$$\Pr[\mathcal{A} \text{ returns 1 in } G_2] = \Pr\left[1 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},1}^{\text{Hiding}}(\lambda)\right].$$

To prove indistinguishability between G_1 and G_2 , we consider a sequence of games where we use the hybrid argument [32] and in which we replace for each $i \in \llbracket \ell_2 \rrbracket$, $A_{(i,2)} = f_{i,m_0[i]}^x$ by $f_{i,m_1[i]}^x$.

Game $G_{1,k}$ (for all $k \in \llbracket \ell_2 \rrbracket$): We define $G_{1,0}$ as G_1 and G_{1,ℓ_2} as G_2 . This game is the same as the game $G_{1,k-1}$ except that the challenger replaces $A_{(k,2)} = f_{k,m_0[k]}^x$ by $f_{k,m_1[k]}^x$ in the commitment, we claim that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1}] - \Pr[\mathcal{A} \text{ returns 1 in } G_{1,k}]| \\ & \leq \frac{1}{2} \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

We prove this claim by reduction, we build the following algorithm \mathcal{B} against the LOR-DDH experiment:

Algorithm $\mathcal{B}(g^x, g^{y_0}, g^{y_1}, g^z)$: generates a multiplicative group $\mathbb{G} = \langle g \rangle$ of prime order p , $\mathcal{M} \leftarrow \{0, 1\}^{\ell_2}$ and $\Theta \leftarrow \{0, 1\}^{\ell_1} \times \{0, 1\}^{\ell_2}$, picks $(u_{i,j})_{\substack{i \in \llbracket \ell_1 \rrbracket \\ j \in \{0,1\}}} \xleftarrow{\$} (\mathbb{Z}_p^*)^{\ell_1}$ and $(v_{i,j}, w_{i,j})_{\substack{i \in \llbracket \ell_2 \rrbracket \\ j \in \{0,1\}}} \xleftarrow{\$} ((\mathbb{Z}_p^*)^{\ell_2})^2$. It sets

set = $(\mathbb{G}, p, (g^{u_{i,j}})_{\substack{i \in \llbracket \ell_1 \rrbracket \\ j \in \{0,1\}}}, (g^{v_{i,j}}, g^{w_{i,j}})_{i \in \llbracket \ell_2 \rrbracket}), \mathcal{M}, \Theta)$ except that it replaces $g^{v_{k,0}}$ by g^{y_0} and $g^{v_{k,1}}$ by g^{y_1} . \mathcal{B} generates $\theta \xleftarrow{\$} \Theta$, parses θ as (s, t) , and runs $(m_0, m_1) \leftarrow \mathcal{A}_1(\text{set})$. \mathcal{B} sets $c = (g^x, ((g^x)^{u_{i,s[i]}})_{i \in \llbracket \ell_1 \rrbracket}, ((g^x)^{v_{i,m_0[i]}}, (g^x)^{w_{i,t[i]}}, (g^x)^{w_{i,1-t[i]}})_{i \in \llbracket \ell_2 \rrbracket})$ except that for any $i < k$, replaces $(g^x)^{v_{i,m_0[i]}}$ by $(g^x)^{v_{i,m_1[i]}}$, and replaces $(g^x)^{v_{k,m_0[k]}}$ by g^z . \mathcal{B} generates a simulated zero-knowledge proof π_* and runs $b' \leftarrow \mathcal{A}(\text{set}, c, \pi_*)$.

- If $m_0[k] = 0$ and $m_1[k] = 1$:
 - If $b' = 0$, then $g^z = g^{x \cdot y_{m_0[k]}} = g^{x \cdot y_0}$.
 - If $b' = 1$, then $g^z = g^{x \cdot y_{m_1[k]}} = g^{x \cdot y_1}$. \mathcal{B} returns b' .
- Else, ($m_0[k] = 1$ and $m_1[k] = 0$):
 - If $b' = 0$, then $g^z = g^{x \cdot y_{m_0[k]}} = g^{x \cdot y_1}$.
 - If $b' = 1$, then $g^z = g^{x \cdot y_{m_1[k]}} = g^{x \cdot y_0}$. \mathcal{B} returns $1 - b'$.

We have:

- 1) $\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1} | m_0[k] = m_1[k]] = \Pr[\mathcal{A} \text{ returns 1 in } G_{1,k} | m_0[k] = m_1[k]]$.
- 2) $\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1} | m_0[k] = 0 \wedge m_1[k] = 1] = \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right]$, and $\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k} | m_0[k] = 0 \wedge m_1[k] = 1] = \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right]$.

This leads to:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1} | m_0[k] = 0 \wedge m_1[k] = 1] - \Pr[\mathcal{A} \text{ returns 1 in } G_{1,k} | m_0[k] = 0 \wedge m_1[k] = 1]| \\ & = \left| \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right] - \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right] \right| \\ & \leq \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

- 3) $\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1} | m_0[k] = 1 \wedge m_1[k] = 0] = \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right] = 1 - \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right]$, and $\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k} | m_0[k] = 1 \wedge m_1[k] = 0] = \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right] = 1 - \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right]$.

This leads to:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns 1 in } G_{1,k-1} | m_0[k] = 1 \wedge m_1[k] = 0] - \Pr[\mathcal{A} \text{ returns 1 in } G_{1,k} | m_0[k] = 1 \wedge m_1[k] = 0]| \\ & = \left| \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right] - \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right] \right| \\ & \leq \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

We obtain:

$$\begin{aligned}
& |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1}] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k}]| \\
&= |\Pr[m_0[k] = m_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | m_0[k] = m_1[k]] + \\
&\quad \Pr[m_0[k] \neq m_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | m_0[k] \neq m_1[k]] - \\
&\quad (\Pr[m_0[k] = m_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | m_0[k] = m_1[k]] + \\
&\quad \Pr[m_0[k] \neq m_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | m_0[k] \neq m_1[k]]) \\
&= \frac{1}{2} \cdot |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | m_0[k] \neq m_1[k]] - \\
&\quad \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | m_0[k] \neq m_1[k]]| \\
&\leq \frac{1}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda).
\end{aligned}$$

Thus, we get:

$$\begin{aligned}
& |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_1] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_2]| \\
&\leq \frac{\ell_2}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda).
\end{aligned}$$

Finally, we have:

$$\begin{aligned}
& \left| \Pr \left[0 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},0}^{\text{Hiding}}(\lambda) \right] - \right. \\
& \left. \Pr \left[1 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},1}^{\text{Hiding}}(\lambda) \right] \right| \\
&= |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_0] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_2]| \\
&\leq \frac{\ell_2}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda),
\end{aligned}$$

which concludes the proof of the lemma since $\frac{\ell_2}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda)$ is negligible. \square

D.2. Proof of Lemma 6 for ORRC

An adversary \mathcal{A} wins the ROS-LDP-Hiding experiment by distinguishing between a message outputted by opening the commitment and a message from the distribution \mathcal{D} (the same distribution as the distribution given by LDP mechanism in Section 3). In this case, we set $n_1 = 2^{\ell_1}$ and $n_2 = 2^{\ell_2}$, the LDP mechanism defines that the probability of having the same message is $\frac{2^{\ell_1+2\ell_2-1}}{2^{\ell_1+\ell_2}}$ and the probability of having other messages is $\frac{2^{\ell_1-1}}{2^{\ell_1+\ell_2}}$. We define the algorithm LDP which output a value picks in the distribution \mathcal{D} as follows:

LDP(m): picks $s, t \xleftarrow{\$} \{0,1\}^{\ell_1} \times \{0,1\}^{\ell_2}$, if $s = 0$, sets $\hat{m} = m$, else, sets $\hat{m} = t$ and returns \hat{m} .

Sim(\hat{m}): simulates the proof $\hat{\pi}$ and returns $\hat{\pi}$.

To prove the lemma, we set the first game as the ROS-LDP-Hiding experiment in the case where $b = 0$ and the last game as the ROS-LDP-Hiding experiment in the case where $b = 1$.

Proof. **Game G_0 :** This game is the same as the ROS-LDP-Hiding experiment in the case $b = 0$, we have:

$$\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_0] = \Pr \left[0 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},0}^{\text{ROS-LDP-Hiding}}(\lambda) \right].$$

Game G_1 : This game is the same as the game G_0 except that the challenger replaces the proofs $\hat{\pi}$ and π_* by simulated proofs, we have:

$$\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_1] = \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_0].$$

From the game G_1 , we parse θ as (s_0, t_0) , the input of the adversary c as $(y, (A_{(i,1)})_{i \in [\ell_1]}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in [\ell_2]})$ and $\hat{\theta}$ as (\hat{s}, \hat{t}) . We note that if $s_0 = \hat{s}$, then $\hat{m} = m$, else $\hat{m} = t_0 \oplus \hat{t}$. (s_0, t_0) being unknown to the adversary, the information of the i -th bit of the message \hat{m}_0 in the commitment c is contained in the i -th bit of the secrets s_0 and t_0 which is hidden in $A_{(i,1)}$ and, respectively $B_{(i,0)}$ and $B_{(i,1)}$.

Game G_2 : This game is the same as the game G_1 except that the challenger replaces for all $i \in [\ell_1]$, $A_{i,1} = g_{i,s_0}^x$ by g_{i,s_1}^x .

To prove indistinguishability, we pick $(s_1, t_1) \xleftarrow{\$} \Theta$ and between G_1 and G_2 , we consider a sequence of games where we use the hybrid argument [32] and in which we replace for each $i \in [\ell_1]$, $A_{(i,1)} = g_{i,s_0}^x$ by g_{i,s_1}^x in the commitment.

Game $G_{1,k}$ (for all $k \in [\ell_1]$): We define $G_{1,0}$ as G_1 and G_{1,ℓ_1} as G_2 . This game is the same as the game $G_{1,k-1}$ except that the challenger replaces $A_{(k,1)} = g_{k,s_0}^x$ by g_{k,s_1}^x , we claim that:

$$\begin{aligned}
& |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1}] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k}]| \\
&\leq \frac{1}{2} \epsilon_{\text{LOR-DDH}}(\lambda).
\end{aligned}$$

We prove this claim by reduction. We build the following PPT algorithm against the LOR-DDH experiment:

Algorithm \mathcal{B} ($g^x, g^{y_0}, g^{y_1}, g^{x \cdot y_b}$): generates a multiplicative group $\mathbb{G} = \langle g \rangle$ of prime order p , $\mathcal{M} \leftarrow \{0,1\}^{\ell_2}$, $\Theta \leftarrow \{0,1\}^{\ell_1} \times \{0,1\}^{\ell_2}$ and $\theta \xleftarrow{\$} \Theta$, parses θ as (s_0, t_0) , picks $(u_{i,j})_{\substack{i \in [\ell_1] \\ j \in \{0,1\}}}$ and $(v_{i,j}, w_{i,j})_{\substack{i \in [\ell_2] \\ j \in \{0,1\}}}$ $\xleftarrow{\$} ((\mathbb{Z}_p^*)^2)^{\ell_2}$. It sets $\text{set} = (\mathbb{G}, p, (g^{u_{i,j}})_{\substack{i \in [\ell_1] \\ j \in \{0,1\}}}, (g^{v_{i,j}}, g^{w_{i,j}})_{\substack{i \in [\ell_2] \\ j \in \{0,1\}}}, \mathcal{M}, \Theta)$ except that it replaces $g^{u_{k,0}}$ by g^{y_0} and $g^{u_{k,1}}$ by g^{y_1} . Runs $r \leftarrow \mathcal{A}_1(\text{set})$ and generates a tuple $c = (g^x, ((g^x)^{u_{i,s_0[i]}})_{i \in [\ell_1]}, ((g^x)^{v_{i,m[i]}}), (g^x)^{w_{i,t_0[i]}}), (g^x)^{w_{i,1-t_0[i]}})_{i \in [\ell_2]})$ (as in the game G_0). Picks $(s_1, t_1) \xleftarrow{\$} \theta$, for any $i < k$, replaces $(g^x)^{u_{i,s_0[i]}}$ by $(g^x)^{u_{i,s_1[i]}}$ and it replaces $(g^x)^{u_{k,s_0[k]}}$ by $g^{x \cdot y_b}$. \mathcal{B} generates a simulated proof π_* and runs $\hat{\theta} \leftarrow \mathcal{A}_2(\text{set}, m, c, \pi_*)$. Parses $\hat{\theta}$ as (\hat{s}, \hat{t}) . If $\hat{s} = s_0$, sets $\hat{m} = m$, else, sets $\hat{m} = t_0 \oplus \hat{t}$. \mathcal{B} generates a simulated proof $\hat{\pi}$ and runs $b' \leftarrow \mathcal{A}_3(\text{set}, m, c, \pi_*, \hat{\theta}, \hat{m}, \hat{\pi})$.

- If $s_0[k] = 0$:
 - If $b' = 0$, then $g^{x \cdot y_b} = g^{x \cdot y_{s_0[k]}} = g^{x \cdot y_0}$.
 - Else, ($b' = 1$), then $g^{x \cdot y_b} = g^{x \cdot y_{s_1[k]}} = g^{x \cdot y_1}$.
- \mathcal{B} returns b' .
- Else, ($s_0[k] = 1$):
 - If $b' = 0$, then $g^{x \cdot y_b} = g^{x \cdot y_{s_0[k]}} = g^{x \cdot y_1}$.
 - Else, ($b' = 1$), then $g^{x \cdot y_b} = g^{x \cdot y_{s_1[k]}} = g^{x \cdot y_0}$.
- \mathcal{B} returns $1 - b'$.

We have:

- 1) $\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] = s_1[k]] = \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] = s_1[k]]$.
- 2) $\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] = 0 \wedge s_1[k] = 1] = \Pr \left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda) \right]$, and

$$\begin{aligned} & \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] = 0 \wedge s_1[k] = 1] \\ &= \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right]. \end{aligned}$$

This leads to:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] = 0 \wedge s_1[k] = 1] - \\ & \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] = 0 \wedge s_1[k] = 1]| \\ &= \left| \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right] - \right. \\ & \quad \left. \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right] \right| \\ &\leq \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

$$\begin{aligned} 3) & \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] = 1 \wedge s_1[k] = 0] \\ &= \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right] \\ &= 1 - \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right], \\ &\text{and} \\ & \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] = 1 \wedge s_1[k] = 0] \\ &= \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right] \\ &= 1 - \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right]. \end{aligned}$$

This leads to:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] = 1 \wedge s_1[k] = 0] - \\ & \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] = 1 \wedge s_1[k] = 0]| \\ &= \left| \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}}(\lambda)\right] - \right. \\ & \quad \left. \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}}(\lambda)\right] \right| \\ &\leq \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

We obtain:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1}] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k}]| \\ &= |\Pr[s_0[k] = s_1[k]] \\ & \quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] = s_1[k]] + \\ & \quad \Pr[s_0[k] \neq s_1[k]] \\ & \quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] \neq s_1[k]] - \\ & \quad (\Pr[s_0[k] = s_1[k]] \\ & \quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] = s_1[k]] + \\ & \quad \Pr[s_0[k] \neq s_1[k]] \\ & \quad \cdot \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] \neq s_1[k]])| \\ &= \frac{1}{2} |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k-1} | s_0[k] \neq s_1[k]] - \\ & \quad \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{1,k} | s_0[k] \neq s_1[k]]| \\ &\leq \frac{1}{2} \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

Thus, we deduce that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_1] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_2]| \\ &\leq \frac{\epsilon_1}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda). \end{aligned}$$

Game G_3 : This game is the same as the game G_2 except that the challenger replaces for all $i \in \llbracket \ell_2 \rrbracket$, $(B_{(i,0)}, B_{(i,1)}) = (h_{i,t_0[i]}^x, h_{i,1 \oplus t_0[i]}^x)$ by $(h_{i,t_1[i]}^x, h_{i,1 \oplus t_1[i]}^x)$, we have:

$$\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_3] = \Pr\left[1 \leftarrow \text{Exp}_{\text{ORFC,LDP},\mathcal{A},1}^{\text{ROS-LDP-Hiding}}(\lambda)\right].$$

To prove indistinguishability between G_2 and G_3 , we consider a sequence of games where we use the

hybrid argument [32] and in which we replace for each $i \in \llbracket \ell_2 \rrbracket$, $(B_{(i,0)}, B_{(i,1)}) = (h_{i,t_0[i]}^x, h_{i,1 \oplus t_0[i]}^x)$ by $(h_{i,t_1[i]}^x, h_{i,1 \oplus t_1[i]}^x)$.

Game $G_{2,k}$ for all $k \in \llbracket \ell_2 \rrbracket$: We define $G_{2,0}$ as G_2 and G_{2,ℓ_2} as G_3 . This game is the same as the game $G_{2,k-1}$ except that the challenger replaces $(B_{(k,0)}, B_{(k,1)}) = (h_{k,t_0[k]}^x, h_{k,1 \oplus t_0[k]}^x)$ by $(h_{k,t_1[k]}^x, h_{k,1 \oplus t_1[k]}^x)$, we claim that:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k-1}] - \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k}]| \\ &\leq \frac{1}{2} \epsilon_{\text{LOR-DDH}-2}(\lambda). \end{aligned}$$

We prove this claim by reduction. We build the following PPT algorithm against the LOR-DDH-2 experiment:

Algorithm $\mathcal{B}(g^x, g^{y_0}, g^{y_1}, g^{x \cdot y_b}, g^{x \cdot y_{1-b}})$: generates a multiplicative group $\mathbb{G} = \langle g \rangle$ of prime order p , $\mathcal{M} \leftarrow \{0, 1\}^{\ell_2}$, $\Theta \leftarrow \{0, 1\}^{\ell_1} \times \{0, 1\}^{\ell_2}$ and $\theta \stackrel{\$}{\leftarrow} \Theta$. Parses θ as (s_0, t_0) , picks $(u_{i,j})_{\substack{i \in \llbracket \ell_1 \rrbracket \\ j \in \{0,1\}}}$ and $(v_{i,j}, w_{i,j})_{\substack{i \in \llbracket \ell_2 \rrbracket \\ j \in \{0,1\}}}$ $\stackrel{\$}{\leftarrow} ((\mathbb{Z}_p^*)^2)^l$. It sets $\text{set} = (\mathbb{G}, p, (g^{u_{i,j}})_{\substack{i \in \llbracket \ell_1 \rrbracket \\ j \in \{0,1\}}}, (g^{v_{i,j}}, g^{w_{i,j}})_{\substack{i \in \llbracket \ell_2 \rrbracket \\ j \in \{0,1\}}}, \mathcal{M}, \Theta)$ except that it replaces $g^{w_{k,0}}$ by g^{y_0} and $g^{w_{k,1}}$ by g^{y_1} . Runs $m \leftarrow \mathcal{A}_1(\text{set})$, picks $(s_1, t_1) \stackrel{\$}{\leftarrow} \theta$, and generates a tuple $\mathbf{c} = (g^x, ((g^x)^{u_{i,s_1[i]}})_{i \in \llbracket \ell_1 \rrbracket}, ((g^x)^{v_{i,m[i]}}, (g^x)^{w_{i,t_0[i]}})_{i \in \llbracket \ell_2 \rrbracket})$ (as in the game G_1). For any $i < k$, replaces $(g^x)^{w_{i,t_0[i]}}$ by $(g^x)^{w_{i,t_1[i]}}$, $(g^x)^{w_{i,1 \oplus t_0[i]}}$ by $(g^x)^{w_{i,1 \oplus t_1[i]}}$ and it replaces $(g^x)^{w_{k,t_0[k]}}$ by $g^{x \cdot y_b}$ and $(g^x)^{w_{k,1 \oplus t_0[k]}}$ by $g^{x \cdot y_{1-b}}$. \mathcal{B} generates a simulated proof π_* and runs $\hat{\theta} \leftarrow \mathcal{A}_2(\text{set}, m, \mathbf{c}, \pi_*)$. Parses $\hat{\theta}$ as (\hat{s}, \hat{t}) .

- If $\hat{s} = s_0$, sets $\hat{m} = m$.
- Else, sets $\hat{m} = t_0 \oplus \hat{t}$.

\mathcal{B} generates a simulated proof $\hat{\pi}$ and runs $b' \leftarrow \mathcal{A}_3(\text{set}, m, \mathbf{c}, \pi_*, \hat{\theta}, \hat{m}, \hat{\pi})$.

- If $t_0[k] = 0$:
 - If $b' = 0$, then $g^{x \cdot y_b} = g^{x \cdot y_{t_0[k]}} = g^{x \cdot y_0}$.
 - If $b' = 1$, then $g^{x \cdot y_b} = g^{x \cdot y_{1 \oplus t_0[k]}} = g^{x \cdot y_1}$.
 - Else ($t_0[k] = 1$):
 - If $b' = 0$, then $g^{x \cdot y_b} = g^{x \cdot y_{t_0[k]}} = g^{x \cdot y_1}$.
 - If $b' = 1$, then $= g^{x \cdot y_{1 \oplus t_0[k]}} = g^{x \cdot y_0}$.
- \mathcal{B} returns b' .

We have:

- 1) $\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k-1} | t_0[k] = t_1[k]] = \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k} | t_0[k] = t_1[k]]$.
- 2) $\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k-1} | t_0[k] = 0 \wedge t_1[k] = 1] = \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}-2}(\lambda)\right]$, and $\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k} | t_0[k] = 0 \wedge t_1[k] = 1] = \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}-2}(\lambda)\right]$.

This lead to:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k-1} | t_0[k] = 0 \wedge t_1[k] = 1] - \\ & \Pr[\mathcal{A} \text{ returns } 1 \text{ in } G_{2,k} | t_0[k] = 0 \wedge t_1[k] = 1]| \\ &= \left| \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH}-2}(\lambda)\right] - \right. \\ & \quad \left. \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH}-2}(\lambda)\right] \right| \\ &\leq \epsilon_{\text{LOR-DDH}-2}(\lambda). \end{aligned}$$

$$\begin{aligned}
3) & \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] = 1 \wedge t_1[k] = 0] \\
&= \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH-2}}(\lambda)\right] \\
&= 1 - \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH-2}}(\lambda)\right], \\
&\text{and} \\
&\Pr[\mathcal{A} \text{ returns 1 in } G_{2,k} | t_0[k] = 1 \wedge t_1[k] = 0] \\
&= \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH-2}}(\lambda)\right] \\
&= 1 - \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH-2}}(\lambda)\right].
\end{aligned}$$

This leads to:

$$\begin{aligned}
&|\Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] = 1 \wedge t_1[k] = 0] - \\
&\Pr[\mathcal{A} \text{ returns 1 in } G_{2,k} | t_0[k] = 1 \wedge t_1[k] = 0]| \\
&= \left| \Pr\left[1 \leftarrow \text{Exp}_{\mathcal{B},1}^{\text{LOR-DDH-2}}(\lambda)\right] - \right. \\
&\quad \left. \Pr\left[0 \leftarrow \text{Exp}_{\mathcal{B},0}^{\text{LOR-DDH-2}}(\lambda)\right] \right| \\
&\leq \epsilon_{\text{LOR-DDH-2}}(\lambda).
\end{aligned}$$

We obtain:

$$\begin{aligned}
&|\Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1}] - \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k}]| \\
&= |\Pr[t_0[k] = t_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] = t_1[k]] + \\
&\quad \Pr[t_0[k] \neq t_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] \neq t_1[k]] - \\
&\quad (\Pr[t_0[k] = t_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] = t_1[k]] + \\
&\quad \Pr[t_0[k] \neq t_1[k]] \\
&\quad \cdot \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] \neq t_1[k]])| \\
&= \frac{1}{2} \cdot |\Pr[\mathcal{A} \text{ returns 1 in } G_{2,k-1} | t_0[k] \neq t_1[k]] - \\
&\quad \Pr[\mathcal{A} \text{ returns 1 in } G_{2,k} | t_0[k] \neq t_1[k]|] \\
&\leq \frac{1}{2} \cdot \epsilon_{\text{LOR-DDH-2}}(\lambda).
\end{aligned}$$

Thus, we deduce that:

$$\begin{aligned}
&|\Pr[\mathcal{A} \text{ returns 1 in } G_2] - \Pr[\mathcal{A} \text{ returns 1 in } G_3]| \\
&\leq \frac{\ell_2}{2} \cdot \epsilon_{\text{LOR-DDH-2}}(\lambda).
\end{aligned}$$

Finally, we have:

$$\begin{aligned}
&\left| \Pr\left[0 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},0}^{\text{ROS-LDP-Hiding}}(\lambda)\right] - \right. \\
&\quad \left. \Pr\left[1 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A},1}^{\text{ROS-LDP-Hiding}}(\lambda)\right] \right| \\
&= |\Pr[\mathcal{A} \text{ returns 1 in } G_0] - \Pr[\mathcal{A} \text{ returns 1 in } G_3]| \\
&\leq \frac{\ell_1}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda) + \frac{\ell_2}{2} \cdot \epsilon_{\text{LOR-DDH-2}}(\lambda),
\end{aligned}$$

which concludes the proof of the lemma since $\frac{\ell_1}{2} \cdot \epsilon_{\text{LOR-DDH}}(\lambda)$ and $\frac{\ell_2}{2} \cdot \epsilon_{\text{LOR-DDH-2}}(\lambda)$ are negligible. \square

D.3. Proof of Lemma 7 for ORRC

An adversary \mathcal{A} wins the Binding experiment when it can open the commitment in two different ways by using the Open algorithm. To prove the lemma, we use the following sequence of games [27] based on failure events, where the first game is the Binding experiment and the last game is the experiment in which the adversary has no chance of winning.

Proof. Game G_0 : This game is the same as the Binding experiment, we have:

$$\Pr[\mathcal{A} \text{ wins the game } G_0] = \Pr\left[1 \leftarrow \text{Exp}_{\text{ORRC,LDP},\mathcal{A}}^{\text{Binding}}(\lambda)\right].$$

Game G_1 : This game is the same as the game G_0 except that the challenger uses the extractors on the zero-knowledge proofs π_0 and π_1 outputted by the adversary and aborts and returns 0 on the event $F_1 =$ ‘‘An extractor fails on at least one proof’’.

We note \mathcal{C} the challenger in the game G_1 . \mathcal{C} extracts the witness x from the proofs π_0 and π_1 . We emphasize that if \mathcal{A} wins its game, then the two witnesses are correctly extracted and these witnesses are the same because the proofs implicitly use the same $y = g^x$. \mathcal{C} parses c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$ and computes $A_2 = \prod_{i=1}^{\ell_2} A_{(i,2)}$, $\alpha_{2,0} = \prod_{i=1}^{\ell_2} f_{i,m_0[i]}$ and

$$\alpha_{2,1} = \prod_{i=1}^{\ell_2} f_{i,m_1[i]}.$$

If the proofs are correctly extracted, we have:

$$y = g^x \wedge A_2 = \alpha_{2,0}^x \text{ and } y = g^x \wedge A_2 = \alpha_{2,1}^x.$$

Let $\epsilon_{\text{ext}}(\lambda)$ be the maximum of failure probability of the extractors, we have:

$$\begin{aligned}
&|\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\
&\leq \Pr[F_1] \leq 2 \cdot \epsilon_{\text{ext}}(\lambda).
\end{aligned}$$

From the game G_1 , an adversary can win the experiment by returning $(m_0, m_1, \pi_0, \pi_1, c)$ such that $m_0 \neq m_1$ and $\prod_{i=1}^{\ell_2} f_{i,m_0[i]} = \prod_{i=1}^{\ell_2} f_{i,m_1[i]}$.

Game G_2 : This game is the same as the game G_1 except that aborts and returns 0 on the event $F_2 =$ ‘‘ \mathcal{A} outputs $(m_0, m_1, \pi_0, \pi_1, c)$ such that $\prod_{i=1}^{\ell_2} f_{i,m_0[i]} = \prod_{i=1}^{\ell_2} f_{i,m_1[i]}$ and $m_0 \neq m_1$ ’’, we have:

$$\begin{aligned}
&|\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\
&\leq \Pr[F_2].
\end{aligned}$$

If the event F_2 occurs, then the adversary \mathcal{A} build $(m_0, m_1, \pi_0, \pi_1, c)$ such that π_0 and π_1 are valid proofs and the adversary wins the experiment.

We claim that:

$$\Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

We prove this claim by reduction. Assume the event F_2 occurs with non negligible probability. We build the following PPT algorithm \mathcal{B} that play the experiment defined in the lemma 4:

Algorithm \mathcal{B} $\left(\left(\tilde{g}_{(i,j)} \right)_{\substack{i \in \llbracket \ell_2 \rrbracket \\ j \in \{0,1\}}} \right)$: simulates the game G_1 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_2 \rrbracket$, it replaces $f_{i,0}$ by $\tilde{g}_{(i,0)}$, and $f_{i,1}$ by $\tilde{g}_{(i,1)}$. Runs $(m_0, m_1, \pi_0, \pi_1, c) \leftarrow \mathcal{A}(\text{set})$. \mathcal{B} computes $\prod_{i=1}^{\ell_2} f_{i,m_0[i]}$ and $\prod_{i=1}^{\ell_2} f_{i,m_1[i]}$, if the event F_2 does not happen *i.e.* if $\prod_{i=1}^{\ell_2} f_{i,m_0[i]} \neq \prod_{i=1}^{\ell_2} f_{i,m_1[i]}$ or $m_0 = m_1$, aborts the experiment, else it returns (m_0, m_1) .

If \mathcal{A} wins the game G_1 and F_2 occurs, then \mathcal{B} returns $(x_1, x_2) \in (\{0, 1\}^{\ell_2})^2$ s.t. $x_1 \neq x_2$ and $\prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_1[i])} = \prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_2[i])}$, which happens with negligible probability $\epsilon_{\ell_2\text{-col}}$ according to the lemma 4.

Thus, we deduce that:

$$|\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \leq \Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

If F_1 and F_2 does not happen, then the values $A_{(i,2)}$ are correct and there is no product collision for two different messages, so in the game G_2 , the adversary has no possibility to win, finally, we have:

$$\begin{aligned} & \Pr\left[1 \leftarrow \text{Exp}_{\text{ORRC, LDP, } \mathcal{A}}^{\text{Binding}}(\lambda)\right] \\ &= |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ &\leq 2 \cdot \epsilon_{\text{ext}}(\lambda) + \epsilon_{\ell_2\text{-col}}(\lambda). \end{aligned}$$

which concludes the proof of the lemma since $2 \cdot \epsilon_{\text{ext}}(\lambda)$ and $2 \cdot \epsilon_{\ell_2\text{-col}}(\lambda)$ are negligible. \square

D.4. Proof of Lemma 8 for ORRC

An adversary \mathcal{A} wins the LDP-Binding experiment by opening the commitment in two different ways using the same seed on OpenLDP algorithm. To prove the lemma, we use the following sequence of games [27] based on failure events, where the first game is the LDP-Binding experiment and the last game is the experiment in which the adversary has no chance of winning.

Proof. Game G_0 : This game is the same as the LDP-Binding experiment, we have:

$$\Pr[\mathcal{A} \text{ wins the game } G_0] = \Pr\left[1 \leftarrow \text{Exp}_{\text{ORRC, LDP, } \mathcal{A}}^{\text{LDP-Binding}}(\lambda)\right].$$

Game G_1 : This game is the same as the game G_0 except that the challenger uses the extractors on the zero-knowledge proofs $\hat{\pi}_0, \hat{\pi}_1$ and π_* outputted by the adversary, and aborts and returns 0 on the event $F_1 =$ ‘‘An extractor fails on at least one proof’’.

We note \mathcal{C} the challenger in the game G_1 . Parses π_* as $(\pi_{A_1}, \pi_{A_2}, \pi_B)$, \mathcal{C} extracts the witnesses from the proofs $\hat{\pi}_0, \hat{\pi}_1, \pi_{A_1}, \pi_{A_2}$ and π_B . We emphasize that if \mathcal{A} wins its game, then the five witnesses are correctly extracted and these witnesses are the same because the proofs implicitly use the same $y = g^x$. \mathcal{C} parses c as $(y, (A_{(i,1)})_{i \in [\ell_1]}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in [\ell_2]})$ and $\hat{\theta}$ as (\hat{s}, \hat{t}) .

$$\begin{aligned} \mathcal{C} \text{ computes } A_2 &= \prod_{i=1}^{\ell_2} A_{(i,2)}, \alpha_{2,0} = \prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]}, \\ \alpha_{2,1} &= \prod_{i=1}^{\ell_2} f_{i, \hat{m}_1[i]}, A_1 = \prod_{i=1}^{\ell_1} A_{(i,1)}, \alpha_1 = \prod_{i=1}^{\ell_1} g_{i, \hat{s}[i]}, \\ \beta_0 &= \prod_{i=1}^{\ell_2} B_{(i, \hat{m}_0[i])}, \beta_1 = \prod_{i=1}^{\ell_2} B_{(i, \hat{m}_1[i])} \text{ and } \gamma = \prod_{i=1}^{\ell_2} h_{i, \hat{t}[i]}. \end{aligned}$$

If the proofs are correctly extracted,

- From the proof $\hat{\pi}_0$, we have:

$$\begin{aligned} & (y = g^x \wedge A_1 = \alpha_1^x \wedge A_2 = \alpha_{2,0}^x) \\ & \vee (y = g^x \wedge A_1 \neq \alpha_1^x \wedge \beta_0 = \gamma^x), \end{aligned}$$

- From the proof $\hat{\pi}_1$, we have:

$$\begin{aligned} & (y = g^x \wedge A_1 = \alpha_1^x \wedge A_2 = \alpha_{2,1}^x) \\ & \vee (y = g^x \wedge A_1 \neq \alpha_1^x \wedge \beta_1 = \gamma^x). \end{aligned}$$

- From the proof π_{A_1} , we have:

$$\bigwedge_{i=1}^{\ell_1} \left(y = g^x \wedge \bigvee_{j=0}^1 A_{(i,1)} = g_{i,j}^x \right),$$

- From the proof π_{A_2} , we have:

$$\bigwedge_{i=1}^{\ell_2} \left(y = g^x \wedge \bigvee_{j=0}^1 A_{(i,2)} = f_{i,j}^x \right),$$

- And from the proof π_B , we have:

$$\bigwedge_{i=1}^{\ell_2} \left(y = g^x \wedge \left(\bigwedge_{j=0}^1 B_{(i,j)} = h_{i,j}^x \vee \bigwedge_{j=0}^1 B_{(i,j)} = h_{i,1-j}^x \right) \right).$$

Let $\epsilon_{\text{ext}}(\lambda)$ be the maximum of failure probability of the extractors, we have:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_0] - \Pr[\mathcal{A} \text{ wins the game } G_1]| \\ & \leq \Pr[F_1] \leq 5 \cdot \epsilon_{\text{ext}}(\lambda). \end{aligned}$$

\hat{s} and $(A_{(i,1)})_{i \in [\ell_1]}$ being given and fixed by the adversary, either we have $A_1 = \alpha_1^x$, otherwise we have $A_1 \neq \alpha_1^x$.

We remark that in the case where we have $A_1 = \alpha_1^x$, an adversary wins the experiment by returning \hat{m}_0 and \hat{m}_1 such that $\pi_*, \hat{\pi}_0$ and $\hat{\pi}_1$ are valid proofs and $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]} =$

$$\prod_{i=1}^{\ell_2} f_{i, \hat{m}_1[i]} \text{ and } \hat{m}_0 \neq \hat{m}_1.$$

In the case where we have $A_1 \neq \alpha_1^x$, an adversary wins the experiment by returning:

- \hat{m}_0 and \hat{m}_1 such that $\pi_*, \hat{\pi}_0$, and $\hat{\pi}_1$ are valid proofs and $\prod_{i=1}^{\ell_2} B_{(i, \hat{m}_0[i])} = \prod_{i=1}^{\ell_2} B_{(i, \hat{m}_1[i])} \left(= \prod_{i=1}^{\ell_2} h_{i, \hat{t}[i]}^x \right)$ and $\hat{m}_0 \neq \hat{m}_1$.
- \hat{m}_0 and \hat{m}_1 such that $\pi_*, \hat{\pi}_0$, and $\hat{\pi}_1$ are valid proofs and $\prod_{i=1}^{\ell_2} h_{(i, \hat{m}_0[i] \oplus \hat{t}[i])} = \prod_{i=1}^{\ell_2} h_{(i, \hat{m}_1[i] \oplus \hat{t}[i])}$ and $\hat{m}_0 \neq \hat{m}_1$.

Game G_2 : This game is the same as the game G_1 except that aborts and returns 0 on the event $F_2 =$ ‘‘ \mathcal{A} returns $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta})$ such that $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]} = \prod_{i=1}^{\ell_2} f_{i, \hat{m}_1[i]}$ and $\hat{m}_0 \neq \hat{m}_1$ ’’, we have:

$$\begin{aligned} & |\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \\ & \leq \Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda). \end{aligned}$$

If the event F_2 occurs, then the adversary \mathcal{A} returns $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta})$ such that $\hat{\pi}_0, \hat{\pi}_1$ and π_* are valid proofs, then the adversary wins the experiment.

We claim that:

$$\Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

We prove this claim by reduction. Assume the event F_2 occurs with non negligible probability, we build the following PPT algorithm \mathcal{B} that play the experiment defined in the lemma 4:

Algorithm \mathcal{B} $\left((\tilde{g}_{(i,j)})_{i \in \llbracket \ell_2 \rrbracket, j \in \{0,1\}} \right)$: simulates the game G_1 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_2 \rrbracket$, it replaces $f_{i,0}$ by $\tilde{g}_{(i,0)}$, and $f_{i,1}$ by $\tilde{g}_{(i,1)}$. Runs $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta}) \leftarrow \mathcal{A}(\text{set})$. \mathcal{B} computes $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]}$ and $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_1[i]}$, if the event F_2 does not happen *i.e.* if $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]} \neq \prod_{i=1}^{\ell_2} f_{i, \hat{m}_1[i]}$ or $\hat{m}_0 = \hat{m}_1$, aborts the experiment, else it returns (\hat{m}_0, \hat{m}_1) .

If \mathcal{A} wins the game G_1 and F_2 occurs, then \mathcal{B} returns $x_1, x_2 \in \{0,1\}^{\ell_2}$ s.t. $x_1 \neq x_2$ and $\prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_1[i])} = \prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_2[i])}$, which happens with non negligible probability $\epsilon_{\ell_2\text{-col}}$ according to the lemma 4.

Thus, we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_1] - \Pr[\mathcal{A} \text{ wins the game } G_2]| \leq \Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

Game G_3 : This game is the same as the game G_2 except that aborts and returns 0 on the event $F_3 = \text{“}\mathcal{A}$ returns $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta})$ such that $\prod_{i=1}^{\ell_2} h_{i, t[i] \oplus \hat{m}_0[i]} = \prod_{i=1}^{\ell_2} h_{i, t[i] \oplus \hat{m}_1[i]}$ and $\hat{m}_0 \neq \hat{m}_1$ and $s \neq \hat{s}$ ”, we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_2] - \Pr[\mathcal{A} \text{ wins the game } G_3]| \leq \Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

If the event F_3 occurs, then the adversary \mathcal{A} returns (\hat{m}_0, \hat{m}_1) such that $\hat{\pi}_0, \hat{\pi}_1$ and π_* are valid proofs and the adversary wins the experiment.

We claim that:

$$\Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

The reduction from the game G_2 to game G_3 is similar as the reduction from the game G_1 to the game G_2 :

Algorithm \mathcal{B} $\left((\tilde{g}_{(i,j)})_{i \in \llbracket \ell_2 \rrbracket, j \in \{0,1\}} \right)$: simulates the game G_2 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_2 \rrbracket$, it replaces $h_{(i,0)}$ by $\tilde{g}_{(i,0)}$, and $h_{(i,1)}$ by $\tilde{g}_{(i,1)}$.

Runs $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta}) \leftarrow \mathcal{A}(\text{set})$. \mathcal{B} uses the extractor on the proof π_* , we note x the witness extracted.

For each $i \in \llbracket \ell_1 \rrbracket$,

- If $A_{(i,1)} = g_{i,0}^x$, sets $s[i] = 0$,
- Else sets $s[i] = 1$.

For each $i \in \llbracket \ell_2 \rrbracket$,

- If $B_{(i,0)} = h_{i,0}^x$, sets $t[i] = 0$,
- Else sets $t[i] = 1$.

\mathcal{B} computes $\prod_{i=1}^{\ell_2} h_{i, t[i] \oplus \hat{m}_0[i]}$ and $\prod_{i=1}^{\ell_2} h_{i, t[i] \oplus \hat{m}_1[i]}$. If the event F_3 does not happen, aborts the experiment, else it

returns $(t \oplus \hat{m}_0, t \oplus \hat{m}_1)$.

If \mathcal{A} wins the game G_2 and F_3 occurs, then \mathcal{B} returns $x_1, x_2 \in \{0,1\}^{\ell_2}$ s.t. $x_1 \neq x_2$ and $\prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_1[i])} = \prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_2[i])}$, which happens with non negligible probability $\epsilon_{\ell_2\text{-col}}$ according to the lemma 4.

Thus, we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_2] - \Pr[\mathcal{A} \text{ wins the game } G_3]| \leq \Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

Game G_4 : This game is the same as the game G_3 except that aborts and returns 0 on the event $F_4 = \text{“}\mathcal{A}$ returns (\hat{m}_0, \hat{m}_1) such that $\prod_{i=1}^{\ell_2} B_{(i, \hat{m}_0[i])} = \prod_{i=1}^{\ell_2} B_{(i, \hat{m}_1[i])}$ and $\hat{m}_0 \neq \hat{m}_1$ and $s \neq \hat{s}$ ”.

If the event F_4 occurs, then the adversary \mathcal{A} returns $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta})$ such that $(\hat{\pi}_0, \hat{\pi}_1)$ are valid proofs, then the adversary wins the experiment.

We claim that:

$$\Pr[F_4] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

The reduction from the game G_3 to game G_4 is similar as the reduction from the game G_1 to the game G_2 :

Algorithm \mathcal{B} $\left((\tilde{g}_{(i,j)})_{i \in \llbracket \ell_2 \rrbracket, j \in \{0,1\}} \right)$: simulates the game G_3 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_2 \rrbracket$, it replaces $h_{i,0}$ by $\tilde{g}_{(i,0)}$, and $h_{i,1}$ by $\tilde{g}_{(i,1)}$. Runs $(\hat{m}_0, \hat{m}_1, \hat{\pi}_0, \hat{\pi}_1, \pi_*, c, \hat{\theta}) \leftarrow \mathcal{A}(\text{set})$. \mathcal{B} computes $\prod_{i=1}^{\ell_2} B_{(i, \hat{m}_0[i])}$ and $\prod_{i=1}^{\ell_2} B_{(i, \hat{m}_1[i])}$. \mathcal{B} uses the extractor on the proof π_* , we note x the witness extracted.

For each $i \in \llbracket \ell_1 \rrbracket$,

- If $A_{(i,1)} = g_{i,0}^x$, sets $s[i] = 0$,
- Else sets $s[i] = 1$.

For each $i \in \llbracket \ell_2 \rrbracket$,

- If $B_{(i,0)} = h_{i,0}^x$, sets $t[i] = 0$,
- Else sets $t[i] = 1$.

If the event F_4 does not happen, aborts the experiment, else it returns $(\hat{m}_0 \oplus t, \hat{m}_1 \oplus t)$.

If \mathcal{A} wins the game G_3 and F_4 occurs, then \mathcal{B} returns $x_1, x_2 \in \{0,1\}^{\ell_2}$ s.t. $x_1 \neq x_2$ and $\prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_1[i])} = \prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_2[i])}$, which happens with non negligible probability $\epsilon_{\ell_2\text{-col}}$ according to the lemma 4.

Thus, we have:

$$|\Pr[\mathcal{A} \text{ wins the game } G_3] - \Pr[\mathcal{A} \text{ wins the game } G_4]| \leq \Pr[F_4] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

If F_1, F_2, F_3 and F_4 does not happen, then the commitment is well-formed and there is no product collision for two different messages on the α_2, β and γ , which implies the uniqueness of the open message for (\hat{s}, \hat{t}) ,

so in the game G_4 , the adversary has no possibility of winning, finally, we have:

$$\begin{aligned} & \Pr \left[1 \leftarrow \text{Exp}_{\text{ORRC, LDP, } \mathcal{A}}^{\text{LDP-Binding}}(\lambda) \right] \\ &= |\Pr [\mathcal{A} \text{ wins the game } G_0] - \Pr [\mathcal{A} \text{ wins the game } G_4]| \\ &\leq 5 \cdot \epsilon_{\text{ext}}(\lambda) + 3 \cdot \epsilon_{\ell_2\text{-col}}(\lambda). \end{aligned}$$

which concludes the proof of the lemma since $5 \cdot \epsilon_{\text{ext}}(\lambda)$ and $3 \cdot \epsilon_{\ell_2\text{-col}}(\lambda)$ are negligible. \square

D.5. Proof of lemma 9 for ORRC

The Probabilistic-LDP-binding security is unachieved when the adversary returns an opened message which does not follow the LDP mechanism distribution according to the seed and the original message. To prove the lemma, we use the following sequence of games [27] based on failure events, where the first game is the Prob-LDP-Binding experiment and the last game is the experiment in which the adversary cannot returning a message that does not follow the same distribution as the distribution of the LDP mechanism.

Proof. Game G_0 : This game is the same as the Prob-LDP-Binding experiment in the case where $b = 0$, we have:

$$\begin{aligned} & \forall \hat{m} \in \mathcal{M}, \\ & \Pr [\hat{m} \leftarrow G_0] = \Pr \left[\hat{m} \leftarrow \text{Exp}_{\text{ORRC, LDP, } \mathcal{A}, 0}^{\text{Prob-LDP-Binding}}(\lambda) \right]. \end{aligned}$$

Game G_1 : From the game G_0 , parses the output of \mathcal{A}_1 as (c, π_*, m, π) , and the output of \mathcal{A}_2 as $(\hat{m}_0, \hat{\pi})$. Parses π_* as $(\pi_{A_1}, \pi_{A_2}, \pi_B, \pi)$. This game is the same as the game G_0 except that the challenger uses the extractors on the zero-knowledge proofs $\pi_{A_1}, \pi_{A_2}, \pi_B, \pi$ and $\hat{\pi}$, and aborts and returns 0 on the event $F_1 = \text{“An extractor fails on at least one proof”}$, we have:

$$\begin{aligned} & \forall \hat{m} \in \mathcal{M}, \\ & |\Pr [\hat{m} \leftarrow G_0] - \Pr [\hat{m} \leftarrow G_1]| \leq \Pr [F_1]. \end{aligned}$$

We note \mathcal{C} the challenger in the game G_1 . \mathcal{C} extracts the witnesses from the proofs $\pi_{A_1}, \pi_{A_2}, \pi_B, \pi$ and $\hat{\pi}$. We emphasize that if the proofs are verified, then the witnesses are correctly extracted and these witnesses are the same because the proofs implicitly used the same $y = g^x$.

\mathcal{C} parses c as $(y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$ and \hat{c} as (\hat{s}, \hat{t}) . \mathcal{C} computes $A_2 = \prod_{i=1}^{\ell_2} A_{(i,2)}$, $\alpha_2 = \prod_{i=1}^{\ell_2} f_{i,m[i]}$, $\alpha_2' = \prod_{i=1}^{\ell_2} f_{i,\hat{m}_0[i]}$, $A_1 = \prod_{i=1}^{\ell_1} A_{(i,1)}$, $\alpha_1 = \prod_{i=1}^{\ell_1} g_{i,\hat{s}[i]}$, $\beta = \prod_{i=1}^{\ell_2} B_{(i,\hat{m}_0[i])}$ and $\gamma = \prod_{i=1}^{\ell_2} h_{i,\hat{t}[i]}$.

The challenger \mathcal{C} finds the tuple (s, t) used by the adversary \mathcal{A} in the commitment c as follows:

For each $i \in \llbracket \ell_1 \rrbracket$, and $k \in \llbracket \ell_2 \rrbracket$

- If $A_{(i,1)} = g_{i,0}^x$, sets $s[i] = 0$, else sets $s[i] = 1$.
- If $B_{(k,0)} = h_{k,0}^x$, sets $t[k] = 0$, else sets $t[k] = 1$.

If the proofs are correctly extracted,

- From the proof π_{A_1} , we have:

$$\bigwedge_{i=1}^{\ell_1} \left(y = g^x \wedge \bigvee_{j=0}^{\ell_1} A_{(i,1)} = g_{i,j}^x \right),$$

- From the proof π_{A_2} , we have:

$$\bigwedge_{i=1}^{\ell_2} \left(y = g^x \wedge \bigvee_{j=0}^{\ell_2} A_{(i,2)} = f_{i,j}^x \right),$$

- From the proof π_B , we have:

$$\bigwedge_{i=1}^{\ell_2} \left(y = g^x \wedge \left(\bigwedge_{j=0}^{\ell_2} B_{(i,j)} = h_{i,j}^x \vee \bigwedge_{j=0}^{\ell_2} B_{(i,j)} = h_{i,1-j}^x \right) \right),$$

- From the proof π , we have:

$$y = g^x \wedge A_2 = \alpha_2^x$$

- And from the proof $\hat{\pi}$, we have:

$$\begin{aligned} & (y = g^x \wedge A_1 = \alpha_1^x \wedge A_2 = \alpha_2^x) \\ & \vee (y = g^x \wedge A_1 \neq \alpha_1^x \wedge \beta = \gamma^x). \end{aligned}$$

Let ϵ_{ext} be the maximum on the failure probability of the extractors, we have:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr [\hat{m} \leftarrow G_0] - \Pr [\hat{m} \leftarrow G_1]| \leq \Pr [F_1] \leq 5 \cdot \epsilon_{\text{ext}}(\lambda).$$

An adversary may attempt to biases the LDP mechanism by returning a commitment $c = (y, (A_{(i,1)})_{i \in \llbracket \ell_1 \rrbracket}, (A_{(i,2)}, B_{(i,0)}, B_{(i,1)})_{i \in \llbracket \ell_2 \rrbracket})$, zero-knowledge proofs π_* and $\hat{\pi}$, and the messages m and \hat{m}_0 such that:

- $\prod_{i=1}^{\ell_1} A_{(i,1)} = \prod_{i=1}^{\ell_1} g_{i,s[i]}^x = \prod_{i=1}^{\ell_1} g_{i,\hat{s}[i]}^x$ and $s \neq \hat{s}$.
- $\prod_{i=1}^{\ell_2} f_{i,m[i]} = \prod_{i=1}^{\ell_2} f_{i,\hat{m}_0[i]}$ and $\hat{m}_0 \neq m$.
- $\prod_{i=1}^{\ell_2} h_{i,\hat{m}_0[i] \oplus t[i]} = \prod_{i=1}^{\ell_2} h_{i,\hat{t}[i]}$ and $\hat{m}_0 \oplus t \neq \hat{t}$.

Game G_2 : This game is the same as the game G_1 except that the challenger aborts and returns 0 on the event $F_2 = \text{“The algorithm } \mathcal{A}_1 \text{ returns } (c, \pi_*, m, \pi) \text{ and } \mathcal{A}_2 \text{ returns } (\hat{m}_0, \hat{\pi}) \text{ such that } \prod_{i=1}^{\ell_2} f_{i,\hat{m}_0[i]} = \prod_{i=1}^{\ell_2} f_{i,m[i]} \text{ and } \hat{m}_0 \neq m\text{”}$, we have:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr [\hat{m} \leftarrow G_1] - \Pr [\hat{m} \leftarrow G_2]| \leq \Pr [F_2].$$

We claim that:

$$\Pr [F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

We prove this claim by reduction.

Assume the event F_2 occurs with non negligible probability, we build the following PPT algorithm \mathcal{B} that play the experiment defined in the lemma 4:

Algorithm \mathcal{B} $\left((\tilde{g}_{(i,j)})_{i \in \llbracket \ell_2 \rrbracket, j \in \{0,1\}} \right)$: simulates the game G_2 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_2 \rrbracket$, it replaces $f_{i,0}$ by $\tilde{g}_{(i,0)}$, and $f_{i,1}$ by $\tilde{g}_{(i,1)}$. Runs $(c, \pi_*, m, \pi) \leftarrow \mathcal{A}_1(\text{set})$, picks $\hat{\theta} \xleftarrow{\$} \Theta$ and runs $(\hat{m}_0, \hat{\pi}) \leftarrow \mathcal{A}_2(\hat{\theta})$. \mathcal{B} computes $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]}$ and $\prod_{i=1}^{\ell_2} f_{i, m[i]}$, if the event F_2 does not happen *i.e.* if $\prod_{i=1}^{\ell_2} f_{i, \hat{m}_0[i]} \neq \prod_{i=1}^{\ell_2} f_{i, m[i]}$ or $\hat{m}_0 = m$, aborts the experiment, else it returns (\hat{m}_0, m) .

If the experiment returns \hat{m}_0 in G_1 and F_2 happens, then \mathcal{B} returns $x_1, x_2 \in \{0,1\}^{\ell_2}$ s.t. $x_1 \neq x_2$ and $\prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_1[i])} = \prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_2[i])}$, which happens with non negligible probability $\epsilon_{\ell_2\text{-col}}$ according to the lemma 4.

Thus, the following holds:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr[\hat{m} \leftarrow G_1] - \Pr[\hat{m} \leftarrow G_2]| \leq \Pr[F_2] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

Game G_3 : Note that in the case where $\hat{s} \neq s$, we have $\hat{m} = t \oplus \hat{t}$, which implies $h_{i, \hat{t}[i]}^x = h_{i, \hat{m}[i] \oplus t[i]}^x$ and $B_{(i, \hat{m}[i])} = h_{i, \hat{t}[i]}^x$. This game is the same as the game G_3 except that aborts and returns 0 on the event $F_3 =$ “The algorithm \mathcal{A}_1 returns (c, π_*, m, π) and \mathcal{A}_2 returns $(\hat{m}_0, \hat{\pi})$ such that $\prod_{i=1}^{\ell_2} h_{i, \hat{m}_0[i] \oplus t[i]} = \prod_{i=1}^{\ell_2} h_{i, \hat{t}[i]}$ and $\hat{m}_0 \oplus t \neq \hat{t}$ and $s \neq \hat{s}$ ”, we have:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr[\hat{m} \leftarrow G_2] - \Pr[\hat{m} \leftarrow G_3]| \leq \Pr[F_3].$$

We claim that:

$$\Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

We prove this claim by reduction.

Assume the event F_3 occurs with non negligible probability, we build the following PPT algorithm \mathcal{B} that play the experiment defined in the lemma 4:

Algorithm \mathcal{B} $\left((\tilde{g}_{(i,j)})_{i \in \llbracket \ell_2 \rrbracket, j \in \{0,1\}} \right)$: simulates the game G_2 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_2 \rrbracket$, it replaces $h_{i,0}$ by $\tilde{g}_{(i,0)}$, and $h_{i,1}$ by $\tilde{g}_{(i,1)}$. Runs $(c, \pi_*, m, \pi) \leftarrow \mathcal{A}_1(\text{set})$, picks $\hat{\theta} \xleftarrow{\$} \Theta$ and runs $(\hat{m}_0, \hat{\pi}) \leftarrow \mathcal{A}_2(\hat{\theta})$. \mathcal{B} uses the extractor on the proof π_* , we note x the witness extracted.

For each $i \in \llbracket \ell_2 \rrbracket$,

- If $B_{(i,0)} = h_{i,0}^x$, sets $t[i] = 0$.
- Else sets $t[i] = 1$.

\mathcal{B} computes $\prod_{i=1}^{\ell_2} h_{i, \hat{m}_0[i] \oplus t[i]}$ and $\prod_{i=1}^{\ell_2} h_{i, \hat{t}[i]}$, if the event

F_3 does not happen *i.e.* if $\prod_{i=1}^{\ell_2} h_{i, \hat{m}_0[i] \oplus t[i]} \neq \prod_{i=1}^{\ell_2} h_{i, \hat{t}[i]}$ or $\hat{m}_0 \oplus t = \hat{t}$ or $s = \hat{s}$, aborts the experiment, else it returns $(\hat{m}_0 \oplus t, \hat{t})$.

If the experiment returns \hat{m}_0 in G_2 and F_3 occurs, then \mathcal{B} returns $x_1, x_2 \in \{0,1\}^{\ell_2}$ s.t. $x_1 \neq x_2$ and

$\prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_1[i])} = \prod_{i=1}^{\ell_2} \tilde{g}_{(i, x_2[i])}$, which happens with non negligible probability $\epsilon_{\ell_2\text{-col}}$ according to the lemma 4.

Thus, we have:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr[\hat{m} \leftarrow G_2] - \Pr[\hat{m} \leftarrow G_3]| \leq \Pr[F_3] \leq \epsilon_{\ell_2\text{-col}}(\lambda).$$

Game G_4 : This game is the same as the game G_3 except that aborts and returns 0 on the event $F_4 =$ “The algorithm \mathcal{A}_1 returns (c, π_*, m, π) and the algorithm \mathcal{A}_2 returns $(\hat{m}_0, \hat{\pi})$ such that $\prod_{i=1}^{\ell_1} g_{i, s[i]} = \prod_{i=1}^{\ell_1} g_{i, \hat{s}[i]}$ and $s \neq \hat{s}$ ”, we have:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr[\hat{m} \leftarrow G_3] - \Pr[\hat{m} \leftarrow G_4]| \leq \Pr[F_4].$$

We claim that:

$$\Pr[F_4] \leq \epsilon_{\ell_1\text{-col}}(\lambda).$$

We prove this claim by reduction.

Assume the event F_4 happens with non negligible probability, we build the following PPT algorithm \mathcal{B} that play the experiment defined in the lemma 4:

Algorithm \mathcal{B} $\left((\tilde{g}_{(i,j)})_{i \in \llbracket \ell_1 \rrbracket, j \in \{0,1\}} \right)$: simulates the game G_3 to \mathcal{A} except that during the setup generation, for each $i \in \llbracket \ell_1 \rrbracket$, it replaces $g_{i,0}$ by $\tilde{g}_{(i,0)}$, and $g_{i,1}$ by $\tilde{g}_{(i,1)}$. Runs $(c, \pi_*, m, \pi) \leftarrow \mathcal{A}_1(\text{set})$, picks $\hat{\theta} \xleftarrow{\$} \Theta$ and runs $(\hat{m}_0, \hat{\pi}) \leftarrow \mathcal{A}_2(\hat{\theta})$. \mathcal{B} uses the extractor on the proof π_* , we note x the witness extracted.

For each $i \in \llbracket \ell_1 \rrbracket$,

- If $A_{(i,1)} = g_{i,0}^x$, sets $s[i] = 0$.
- Else sets $s[i] = 1$.

\mathcal{B} computes $\prod_{i=1}^{\ell_1} g_{i, s[i]}$ and $\prod_{i=1}^{\ell_1} g_{i, \hat{s}[i]}$, If the event F_4 does not happen, aborts the experiment, else it returns (s, \hat{s}) .

Thus, we have:

$$\forall \hat{m} \in \mathcal{M},$$

$$|\Pr[\hat{m} \leftarrow G_3] - \Pr[\hat{m} \leftarrow G_4]| \leq \Pr[F_4] \leq \epsilon_{\ell_1\text{-col}}(\lambda).$$

If F_1, F_2, F_3 , and F_4 does not happen, then the commitment is well formed and the only way to open the commitment is to return $\hat{m} = m$ if $s = \hat{s}$, and to return $t \oplus \hat{t}$ otherwise. We deduce that the game G_4 is the same as the Prob-LDP-Binding experiment in the case where $b = 1$.

Finally, we deduce that:

$$\forall \hat{m} \in \mathcal{M},$$

$$\begin{aligned} & \left| \Pr[\hat{m} \leftarrow \text{Exp}_{\text{ORRC, LDP, } \mathcal{A}, 0}^{\text{Prob-LDP-Binding}}(\lambda)] - \right. \\ & \left. \Pr[\hat{m} \leftarrow \text{Exp}_{\text{ORRC, LDP, } \mathcal{A}, 1}^{\text{Prob-LDP-Binding}}(\lambda)] \right| \\ &= |\Pr[\hat{m} \leftarrow G_0] - \Pr[\hat{m} \leftarrow G_4]| \\ &\leq 5 \cdot \epsilon_{\text{ext}}(\lambda) + \epsilon_{\ell_1\text{-col}}(\lambda) + 2 \cdot \epsilon_{\ell_2\text{-col}}(\lambda). \end{aligned}$$

which concludes the proof of the lemma since $5 \cdot \epsilon_{\text{ext}}(\lambda)$, $2 \cdot \epsilon_{\ell_2\text{-col}}(\lambda)$ and $\epsilon_{\ell_1\text{-col}}(\lambda)$ are negligible. \square