Efficient Laconic Oblivious Transfer with Malicious Security*

Yijian Zhang, Zuoxia Yu✉, Rupeng Yang, and Willy Susilo

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong NSW, Australia.
{yijianz,zyu,rupengy,wsusilo}@uow.edu.au

Abstract. Laconic *Oblivious Transfer* (OT) enables a receiver to compress a large database into a short digest, allowing a sender to generate a succinct ciphertext that reveals only a single indexed message. While prior works achieve efficient constructions in the semi-honest model, known solutions for malicious security rely on inherently heavy non-black-box techniques, such as garbled circuits or succinct arguments, resulting in impractical schemes.

In this work, we present the first construction of *maliciously secure laconic OT that makes fully black-box use of its underlying primitives*. Our construction is based on a simple and widely used paradigm combining vector commitments with witness encryption, previously known to yield efficient semi-honest protocols. We show that this approach can in fact achieve malicious security by introducing two minimal yet powerful enhancements: equivocal vector commitments and maliciously secure witness encryption. We prove that these properties suffice to obtain laconic OT with security against malicious senders in the universal composability model and against malicious receivers in the context model.

We instantiate our framework from standard assumptions, including LWE and bilinear Diffie-Hellman variants, obtaining schemes in the standard model. Notably, our constructions avoid heavy non-black-box tools mentioned before, and therefore achieve efficiency comparable to the state-of-the-art black-box constructions with merely semi-honest security. In particular, in certain settings, we obtain malicious security at essentially no additional asymptotic cost.

Our results resolve a central question on the feasibility of practical, maliciously secure laconic OT, and provide a unifying pathway for efficiently achieving malicious security in laconic cryptography.

1 Introduction

An *Oblivious Transfer* (OT) protocol [RAB81,Kil88] enables a receiver with a bit b to learn a message m_b from a sender who holds two messages m_0 and m_1. Its security requirements are twofold: the *sender privacy* requires that the receiver cannot learn any information about m_{1-b}, and the *receiver privacy* requires that the sender cannot learn the bit b.

Laconic OT [CDG+17] enhances standard OT to minimize the computation and communication costs in applications that need a large number of OT instances. In a laconic OT protocol, the receiver holds a large database D instead of a single bit, which will be compressed into a short digest dig. Given this digest, the sender encrypts two messages m_0 and m_1 with an index L. Then the receiver decrypts the ciphertext and obtains $\mathsf{m}_{D[L]}$. For security, the sender privacy and the receiver privacy guarantee the confidentiality of $\mathsf{m}_{1-D[L]}$ and D, respectively. An essential requirement of this primitive is that the size of dig and ct, as well as the runtime of encryption, are only poly-logarithmic in the database size $|D|$. Laconic OT has found numerous applications in practice, especially in the scenarios of secure multi-party computation (MPC) protocols, e.g., laconic private set intersection [ABD+21],

* We appreciate the anonymous reviewers for their valuable comments. Yijian Zhang and Zuoxia Yu are partially supported by the Australian Research Council (ARC) Discovery Early Career Research (DECRA) DE240100282. Rupeng Yang is partially supported by ARC DECRA DE250101041. Willy Susilo is supported by ARC Laureate Fellowship FL230100033.

adaptively secure garbled circuits [GS18a], succinct functional encryption [KNTY19], laconic function evaluation [Ros22,DHMW24], and more broadly, two-round MPC in general [GS17,GS18b].

Laconic OT with Semi-Honest Security. Most constructions of laconic OT are only proven secure in the *semi-honest* model, where both sender and receiver are assumed to be *honest-but-curious*, i.e., they will proceed as regulated by the protocol but remain curious about the input of the other party.

The first construction of laconic OT is proposed by Cho et al. in [CDG+17]. However, the construction compiles public-key cryptographic operations into garbled circuits, which is extremely impractical. This motivated a series of subsequent works [GVW20,GJV23,DKL+23,FHAS24] that attempt to construct more efficient laconic OT without using such heavy non-black-box techniques. These black-box constructions roughly follow a paradigm summarized in [FHAS24]: the receiver compresses the database D into a short commitment using a vector commitment scheme; then the sender uses a witness encryption to encrypt the two messages m_0 and m_1 with an index L, where m_b can be recovered iff the commitment is opened to b at position L. While it is difficult to construct a general witness encryption (WE) [GGSW13], these works have constructed specific witness encryption schemes that work in their settings, including trapdoor hash encryption [DGI+19], one-way functions with encryption (OWFE) [GVW20], set membership encryption [GJV23], laconic encryption [DKL+23], and witness encryption for vector commitment [FHAS24]. Besides, we note that laconic OT can be constructed via the above paradigm using registration-based encryption (RBE) [GHMR18,GKMR23,FKdP23]. These constructions and implications provide efficient constructions of laconic OT from various assumptions. However, all of them will be insecure if the parties cheat.

Laconic OT with Malicious Security. In this work, we focus on laconic OT with the more natural *malicious* security, where the adversary is allowed to behave arbitrarily in the protocol, e.g., the receiver may attempt to learn both messages by generating an ill-formed digest. Many applications of laconic OT require it to have malicious security. For example, Dottling et al. [DGGM19] construct laconic conditional disclosure of secrets from malicious laconic OT, and then use it to construct maliciously secure Bob-optimized two-party computation protocol and laconic function evaluation. In addition, Eldridge et al. [EGG+22] showed that malicious laconic OT can be transformed into one-time programs, which is a powerful cryptographic primitive with many appealing applications.

In order to achieve malicious security, there is a folklore approach based on general-purpose *Succinct Non-Interactive Argument of Knowledge* (SNARK) [BCCT12,BCG+13], which forces the sender and the receiver to act honestly. However, this approach is non-black-box since it needs the sender/receiver to compile their algorithms into circuits and then invoke the prover algorithms of SNARK to prove the satisfiability of the circuits. Even worse, due to the strong negative results from [GW11], security of SNARK must be established either in the random oracle model [Mic94] or in the standard model from some non-falsifiable knowledge assumptions [Gro10]. Döttling et al. [DGGM19] proposed the first construction of malicious laconic OT based on standard assumptions in the standard model, which can be instantiated from either the standard Computational Diffie-Hellman (CDH) assumption or the Decisional Learning with Errors (DLWE) assumption. However, the construction is still non-black-box. First, to achieve receiver privacy against a malicious sender, they will use a general-purpose *Non-Interactive Zero-Knowledge Argument of Knowledge* (NIZK) [FLS90,GS08] to prove the honest behavior of the sender, and this will make non-black-box use of the sender's operations. In addition, their construction contains a large number of garbled circuits compiled from public-key operations, which is also non-black-box and heavy. Alamati et al. [ABD+21] made attempts towards black-box malicious laconic OT by applying an efficient special-purpose NIZK, resulting in a concrete laconic OT protocol secure against malicious senders. However, their protocol does not have sender privacy against malicious receivers, which is much more challenging to achieve.

To recap, existing constructions of laconic OT with full malicious security must rely on impractical non-black-box techniques, and it is natural to ask:

Can we construct a laconic OT with malicious security in a black-box fashion?

1.1 Our Results

In this work, we present a new generic construction of maliciously secure laconic OT, which only makes black-box use of underlying building blocks. Our construction achieves receiver privacy against malicious senders in the *universal composability* (UC) model of Canetti [Can01] and sender privacy against malicious receivers in the *context* model of Döttling et al. [DGGM19][1]. This matches the security achieved by the state-of-the-art constructions of maliciously secure laconic OT [DGGM19].

Following previous constructions of black-box laconic OT (with semi-honest security), we construct the scheme from a vector commitment and an associated witness encryption scheme. However, we notice that previous definitions of both primitives are not sufficient for achieving malicious security. We then enhance the vector commitment to be equivocal and enhance the witness encryption to have malicious security. We prove that malicious laconic OT can be constructed from an equivocal vector commitment and an associated malicious witness encryption.

We then show how to instantiate the underlying building blocks from lattices and pairings in the standard model. In particular, security of our pairing-based instantiations relies on either the Computational Bilinear Diffie-Hellman Exponent (CBDHE) assumption or the Computational Bilinear Diffie-Hellman (CBDH) assumption, and security of our lattice-based instantiation relies on the standard DLWE assumption with a polynomial modulus-to-noise ratio. Since our constructions avoid heavy tools like SNARKs, NIZKs, and garbled circuits, their efficiency is comparable to that of previous black-box laconic OT with semi-honest security. Besides, we also leverage the random oracle model in our constructions to reduce some operations, which improves the efficiency by an order of $O(\lambda)$. In this setting, our pairing-based constructions achieve the same asymptotic efficiency as state-of-the-art constructions [GKMR23,FKdP23] with merely semi-honest security, i.e., malicious security is achieved almost for free.

We give a comprehensive comparison of our protocols and the previous ones in Table 1. To compare the efficiency of all schemes that rely on different mathematical tools, we evaluate the parameter sizes ($|\mathsf{crs}|, |\mathsf{dig}|, |\mathsf{ct}|$) in terms of the security parameter λ and the database size N as follows: for schemes over lattices, the lattice dimension is set to be $O(\lambda)$; for schemes over bilinear groups, each group element is of size $O(\lambda)$; for schemes over integer groups (based on CDH or ϕ-hiding), each group element is of size $O(\lambda^3)$ [LL93,Gen10,BG18][2]. Moreover, for those schemes employing garbled circuits, we assume that the optimal garbled circuit scheme [LGW24][3] is employed and it requires $O(\lambda)$ bits of communication for each gate in the circuit to be garbled.

1.2 Related Work

Laconic Function Evaluation. In laconic function evaluation, the receiver can further compress a large-sized function f into a short digest, regarding which the sender encrypts his input data x to generate a ciphertext. The ciphertext reveals nothing about x except for $f(x)$. Quach et al. [QWW18] constructed the first laconic function evaluation for general circuits from DLWE. Döttling et al. [DGM23] proposed a concrete scheme supporting Turing machines, relying on indistinguishability obfuscation. Dong et al. [DHMW24,DHM+24] proposed their schemes for random-access machines regarding weak efficiency and full efficiency, respectively. Orthogonally, to obtain an

[1] The standard UC model requires the communication costs to scale with the input size of the malicious parties and thus cannot be achieved in this setting. Therefore, [DGGM19] proposed the relaxed context model for laconic protocols. We refer the readers to [DGGM19] for a more detailed discussion on both models.

[2] Note that CDH-based schemes can also be instantiated over more succinct curve-based groups, but this may lead to a larger ct, as these schemes will compile complex curve operations into garbled circuits.

[3] One may consider the recent efficient garbled circuit scheme with 1 bit per gate communication of Liu et al. [LWYY25], while this will only slightly improve the parameter $|\mathsf{ct}|$, and it requires stronger circular assumptions.

Table 1. The comparison of existing laconic OT schemes

Security	Reference	|crs|	|dig|	|ct|	BB	MR	MS	PQ	Model	Assump.
Semi-Honest	[CDG+17]	$O(\lambda^4)$	$O(\lambda^3)$	$O(\lambda^8 \log N)$	×	×	×	×	Stand	CDH
	[GVW20]	$O(\lambda N)$	$O(\lambda)$	$O(\lambda)$	✓	×	×	×	Stand	DBDHI
	[DKL+23]	$O(\lambda^2)$	$O(\lambda)$	$O(\lambda \log N)$	✓	×	×	✓	Stand	DLWE
	[GKMR23]‡	$O(\lambda N)$	$O(\lambda)$	$O(\lambda)$	✓	×	×	×	Stand	DBDHE
		$O(\lambda N^2)$	$O(\lambda)$	$O(\lambda)$	✓	×	×	×	Stand	DBDH
	[FKdP23]‡	$O(\lambda N \log N)$	$O(\lambda \log N)$	$O(\lambda \log N)$	✓	×	×	×	Rand	DBDHE
		$O(\lambda^2 \log N)$	$O(\lambda \log N)$	$O(\log^2 N)$	✓	×	×	✓	Rand	DLWE
	[GJV23]	$O(\lambda N)$	$O(\lambda)$	$O(\lambda)$	✓	×	×	×	GGM	sDBDHE
	[FHAS24]	$O(\lambda N)$	$O(\lambda)$	$O(\lambda)$	✓	×	×	×	AGM	DL
Malicious	[ABD+21]§	$O(\lambda^3)$	$O(\lambda^4)$	$O(\lambda^4)$	✓	×	✓	×	Stand	ϕ-hiding
	[DGGM19]	$O(\lambda^4)$	$O(\lambda^3)$	$O(\lambda^9 \log N)$	×	✓	✓	×	Stand	CDH
		$O(\lambda^4)$	$O(\lambda^2)$	$O(\lambda^6 \log N)$	×	✓	✓	✓	Stand	DLWE
	This work	$O(\lambda N)$	$O(\lambda)$	$O(\lambda^2)/O(\lambda)$	✓	✓	✓	×	Stand/Rand	CBDHE
		$O(\lambda N^2)$	$O(\lambda)$	$O(\lambda^2)/O(\lambda)$	✓	✓	✓	×	Stand/Rand	CBDH
		$O(\lambda^2)$	$O(\lambda)$	$O(\lambda^3 \log N)/O(\lambda^2 \log N)$	✓	✓	✓	✓	Stand/Rand	DLWE

* The columns "BB", "MR", "MS", and "PQ" indicate whether the construction only uses the black-box technique, resists a malicious receiver, resists a malicious sender, and provides the post-quantum security.

** The columns "Model" and "Assump." stand for the security model and assumption, respectively, where Stand, Rand, GGM, and AGM represent the standard model, the random oracle model, the generic group model [Sho97] and the algebraic group model [FKL18]. (C/D)BDH(I/E) represents the (Computational/Decisional) Bilinear Diffie-Hellman (Inversion/Exponent) assumption. Besides, sDBDHE represents the selective DBDHE assumption, which is a stronger variant of the DBDHE assumption.

‡ The semi-honest laconic OT can be derived from recent constructions of registration-based encryption [FKdP23,GKMR23].

§ The construction in [ABD+21] is not fully maliciously secure because it cannot resist a malicious receiver.

optimized efficiency, Hsieh et al. [HLL23] presented the first laconic function evaluation for circuits with parameters independent of circuit depth. The recent techniques proposed by Wee [Wee24,Wee25] achieved more succinct public parameters. Very recently, Meyer et al. [MORS26] proposed the first attribute-based laconic function evaluation from the decisional composite residuosity (DCR) assumption. Their construction, when combined with fully homomorphic encryption, implies laconic function evaluation.

Laconic Private Set Intersection. This primitive achieves lower communication cost at the receiver's side for two-round private set intersection protocols. Alamati et al. [ABD+21] formalized its definition and showed how to construct laconic private set intersection in a non-black-box fashion. They also presented a black-box construction based on the ϕ-hiding assumption. Subsequently, Aranha et al. [ALOS22] presented a black-box scheme over pairings. Bartusek et al. [BGJP25] recently proposed a black-box construction of laconic private set intersection on authenticated inputs that achieves malicious security via the distinguisher-dependent simulation technique [JKKR17]. We note that this technique is also the basis of the context security model.

2 Technical Overview

Definition. We start by recalling the formal definition of laconic OT. This protocol involves two parties: the receiver who holds a database $D \in \{0,1\}^N$ and the sender who specifies a location index $L \in [0, N-1]$ and a pair of messages $(\mathsf{m}_0, \mathsf{m}_1)$. The protocol works in four steps:

– It initializes the protocol by running $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N)$ where λ is the security parameter and crs is a common reference string.

– In the first round, the receiver runs $(\mathsf{dig}, \mathsf{aux}) \leftarrow \mathsf{Hash}(\mathsf{crs}, D)$ and then returns dig to the sender, where dig is a short digest of size $\mathsf{poly}(\log N)$ and aux is auxiliary information.
– In the second round, the sender computes $\mathsf{ct} \leftarrow \mathsf{Send}(\mathsf{crs}, \mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1)$ and returns ct to the receiver.
– With random access to aux, the receiver obtains $\mathsf{m}_{D[L]} \leftarrow \mathsf{Receive}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{ct}, L)$. Without loss of generality, we further assume $\mathsf{aux} = (\mathsf{aux}_0, \ldots, \mathsf{aux}_{N-1})$, where aux_L stores all information the receiver will access. Therefore, this algorithm can be rewritten as $\mathsf{Receive}(\mathsf{crs}, \mathsf{aux}_L, \mathsf{ct}, L)$.

Defining Malicious Security. The malicious security is considered in a *composable* setting. Ideally, it requires that any external context[4] cannot tell whether it is interacting with a real-world malicious adversary or an ideal-world simulator. In the real world, the context invokes the adversary and provides the input to the honest party, and then the parties interact with each other as in laconic OT. Here, the context can communicate with the adversary at any time and acquire all their outputs. However, in the ideal world, the adversary is replaced by a simulator that only interacts with a trusted ideal functionality F, and the honest party also only interacts with F. The laconic OT considers F that takes as inputs $(L, \mathsf{m}_0, \mathsf{m}_1)$ (from the sender side) and D (from the receiver side), then outputs $(L, \mathsf{m}_{D[L]})$ to the receiver. Finally, the context distinguishes whether it is in the ideal world or not.

As a benefit of achieving composability, nothing about $\mathsf{m}_{\overline{D}[L]}$ or D is leaked in any context, where we assume $\overline{D}[L] := 1 - D[L]$. Next, we describe the best-known security model against malicious parties.

When the adversary colludes with a malicious sender. In this case, the simulator will interact with F on behalf of the malicious sender, and simulate sender's view, namely dig (without knowing D), to answer possible queries from the context. The simulator has to submit the message $(L, \mathsf{m}_0, \mathsf{m}_1)$ to F, while it might only see a ciphertext ct (from the context), so it should be able to extract $(\mathsf{m}_0, \mathsf{m}_1)$ from ct[5]. Here, we adopt the UC model [Can01], in which the simulator should be straight-line and does not depend on the context.

When the adversary colludes with a malicious receiver. In this case, the simulator will interact with F on behalf of the malicious receiver, and simulate receiver's view, namely ct (without knowing $\mathsf{m}_{\overline{D}[L]}$), to answer possible queries from the context. Similarly, the simulator needs to extract D from a short digest dig, while this is impossible in an information-theoretic sense. Then we adopt the context model [DGGM19], which exactly resolves this issue.

In a nutshell, the context model assumes a new entity called *extractor* to handle the extraction of D, relying on the distinguisher-dependent simulation technique [JKKR17]. At the core of this technique, the extractor can guess each bit of D by having the black-box access to the distinguisher of the context, where the distinguisher takes as input the sender ciphertext and outputs a bit indicating the real/ideal world. Therefore, given two real messages $(\mathsf{m}_0, \mathsf{m}_1)$ and a random message $\widetilde{\mathsf{m}}$, if we feed encryption of either $(\mathsf{m}_0, \widetilde{\mathsf{m}})$ or $(\widetilde{\mathsf{m}}, \mathsf{m}_1)$ to the distinguisher and then see an obvious difference for the output, the extractor can determine the value of $D[L]$ and hence iteratively obtain the entire D. In this way, the context security can be reduced to a weaker malicious security notion, namely indistinguishability-based (IND) security. It states that for any adversary (colluding with a malicious receiver) there must exist a bit $\beta \in \{0, 1\}$ such that the advantage of distinguishing the encryption of $(\mathsf{m}_0, \mathsf{m}_1)$ from the encryption of $(\mathsf{m}_\beta, \widetilde{\mathsf{m}})$ is negligible, while that of distinguishing the encryption of $(\mathsf{m}_0, \mathsf{m}_1)$ from the encryption of $(\mathsf{m}_{1-\beta}, \widetilde{\mathsf{m}})$ is non-negligible. In the following, we will focus on the IND security against malicious receivers.

2.1 Our Paradigm of Malicious Laconic OT

Recap: WE for Vector Commitment. We will first recall the notion of witness encryption (WE) for vector commitment, which can be effectively constructed from weak assumptions [FKdP23,FHAS24,CFK24,GHK+25].

[4] Prior work also refers to this entity as the environment.
[5] We ignore the input L, as L is assumed to be explicitly revealed in ct.

WE [GGSW13] allows users to generate a ciphertext ct with respect to a message m and a statement s in a predefined language. Anyone who has a valid witness w for s can decrypt ct and thus obtain m. Here, we consider a vector commitment relation language in which the user can compress a binary vector $D \in \{0,1\}^N$ into a succinct commitment com using a common reference string crs_c. Given a statement $s_b = (\mathsf{com}, L, b)$, a valid witness w is a proof π_L of opening a bit b at the L-th location of D. We say that s is not a false statement only if $b = D[L]$.

With above tools, we can easily derive a laconic OT scheme as follows:

$$\mathsf{crs} := \mathsf{crs}_c, \quad \mathsf{dig} := \mathsf{com}, \quad \mathsf{aux}_L := (D[L], \pi_L), \quad \mathsf{ct} := (\mathsf{ct}_0, \mathsf{ct}_1) \tag{1}$$

where ct_b for $b \in \{0,1\}$ is a WE ciphertext generated from a message m_b and a statement $s_b = (\mathsf{com}, L, b)$. Next, we informally analyze the correctness, laconism, and security of the scheme:

- The correctness follows the correctness of WE, i.e., $\mathsf{ct}_{D[L]}$ can be decrypted when the receiver owns $w = \pi_L$.
- The laconism follows the succinctness of vector commitment, i.e., com is of size $\mathsf{poly}(\log N)$.
- The (semi-honest) security roughly follows the hiding property of vector commitment and the standard semantic security of WE.
  - By the hiding property of vector commitment, dig will not leak any information about D, so that we can simulate dig without knowing D.
  - By the semantic security of WE, given the vector D and a false statement $s_{\overline{D}[L]} = (\mathsf{com}, L, \overline{D}[L])$, $\mathsf{ct}_{\overline{D}[L]}$ is indistinguishable from a ciphertext that encrypts a random message. Therefore, it can blind $\mathsf{m}_{\overline{D}[L]}$ in the reduction and the simulation of ct only needs $(L, \mathsf{m}_{D[L]})$.

The Challenges. We remark that the paradigm presented in (1) is actually insufficient to deal with malicious parties. In particular, we analyze the reasons as follows:

(1) When proving the UC security against malicious senders, it is hard to extract $(\mathsf{m}_0, \mathsf{m}_1)$ if we only rely on the black-box use of other building blocks. All prior works generally employ an extractable NIZK to handle this issue, while this method always makes non-black-box use of NIZK. What's worse, not all WE schemes can find an efficient NIZK protocol, such as the lattice-based one in [FKdP23].

(2) When proving the IND security against malicious receivers, confidentiality of $\mathsf{m}_{\overline{D}[L]}$ cannot be reduced to the semantic security of WE. This is mainly because in previous definitions of witness encryption for vector commitment [FKdP23,CFK24,FHAS24], semantic security is only defined in the setting that the commitment is honestly generated; however, in the definition of security of laconic OT, the commitment, which is the digest of the receiver, can be maliciously generated. Besides, definition of WE for vector commitment needs the adversary to provide both D and $s_b = (\mathsf{com}, L, b)$ to check if the statement is false or not; however, in the security definition of laconic OT, only the statement s_b is revealed by the adversary. Therefore, the semantic security of WE is insufficient to resist malicious receivers.

Solution: Malicious WE for Equivocal Vector Commitment. To solve above challenges, we formalize and instantiate a novel variant of WE, namely *malicious WE for equivocal vector commitment*. We distinguish this new primitive from prior WEs in terms of the following aspects:

- *Statistical Equivocality*: A dummy commitment $\widetilde{\mathsf{com}}$ can be generated without knowing the vector D, and it can be later equivocated into a valid opening proof $\widetilde{\pi}_{L,b}$ with respect to any location $L \in [0, N-1]$ and $b \in \{0,1\}$, via the trapdoor information td. It critically requires that $(\widetilde{\mathsf{com}}, \widetilde{\pi}_{L,b})$ is statistically indistinguishable from normally generated (com, π_L) (when $D[L] = b$).
- *Malicious IND security*: The semantic security holds even if the commitment is generated by the adversary maliciously. In addition, given only (com, L) and two messages $(\mathsf{m}_0, \mathsf{m}_1)$, without knowing the vector D, there must exist some $\beta \in \{0,1\}$ such that the WE encryption of m_β is pseudorandom.

Proof Strategy. Now we instantiate the paradigm in (1) with our malicious WE for equivocal vector commitment. We explain how to prove malicious security.

Resisting malicious senders via equivocality. We observe that the extraction of $(\mathsf{m}_0, \mathsf{m}_1)$ can be done via an equivocal vector commitment. In the proof, with a trapdoor td, the simulator can simulate a dummy digest $\widetilde{\mathsf{dig}} := \widetilde{\mathsf{com}}$, and normally decrypt both ct_0 and ct_1 (from the context) using equivocated opening proofs $(\widetilde{\pi}_{L,0}, \widetilde{\pi}_{L,1})$. As these proofs are statistically indistinguishable from normally generated ones and the context is not aware of the usage of $\widetilde{\pi}_{L,\overline{D}[L]}$, $(\mathsf{m}_0, \mathsf{m}_1)$ can be properly recovered via the decryption algorithm of WE and the context cannot observe such difference in the ideal world.

Our equivocality notion stems from the private opening property of vector commitment [WW23], which also supports the simulation of $(\widetilde{\mathsf{com}}, \widetilde{\pi}_{L,b})$ without knowing D, while a major difference is that the commitment should be compatible with a malicious WE scheme and we critically require statistical equivocality to guarantee the proper decryption of WE ciphertexts.

Resisting malicious receivers via malicious IND security. The malicious IND security of WE is defined when the commitment may be maliciously generated. Besides, it holds even when it is unclear if the statement is false or not. This exactly implies the IND security against malicious receivers of laconic OT. Specifically, since the ciphertext of laconic OT consists of two WE ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$, where ct_b encrypts the message m_b under the statement $s_b = (\mathsf{com}, L, b)$, the malicious IND security of WE tells that there exists a bit $\beta \in \{0,1\}$ such that the adversary cannot distinguish the encryption of m_β from that of a random message. Therefore, this must hold when $\beta = \overline{D}[L]$; on the contrary, the advantage of the adversary would be non-negligible when $\beta = D[L]$ as it can always decrypt $\mathsf{ct}_{D[L]}$. This gives a difference for distinguishing the ciphertexts encrypting different messages, so that we can reduce the malicious IND security of WE to the IND security against malicious receivers of laconic OT.

The next question is how to construct a malicious WE scheme. The main challenge is that existing security proofs of WE for vector commitment need the knowledge of D to simulate their ciphertexts. To solve this problem, we refer to the proof strategy of [DGGM19]: starting from weaker one-way (OW) security and then upgrading it to malicious IND security via general search-to-decision techniques. We observe that this strategy works well even when constructing black-box laconic OT. Therefore, we define the malicious OW security as follows:

– *Malicious OW security*: only given (com, L) and two randomly sampled messages $(\mathsf{m}_0, \mathsf{m}_1)$, without knowing the vector D, we can generate ct_b that encrypts m_b under the statement $s_b = (\mathsf{com}, L, b)$ for $b \in \{0,1\}$ such that $(\mathsf{m}_0, \mathsf{m}_1)$ is unpredictable for the adversary.

For some $\beta \in \{0,1\}$, if m_β is unpredictable, then we can further demonstrate its pseudorandomness via either the Goldreich-Levin hardcore predicate [GL89] or the random oracle, so that ct_β will not leak any information about m_β and we can treat m_b as a random one-time key to encrypt the real message of WE, thus achieving the malicious IND security. Note that if adopting the Goldreich-Levin hardcore predicate, a subtle issue is that the above OW security only guarantees the unpredictability of the entire $(\mathsf{m}_0, \mathsf{m}_1)$, rather than m_β. This can be resolved via the standard hardness amplification technique [CHS05], which incurs $O(\lambda)$ larger encryption overheads but ensures the security in the standard model. Next, we will focus on the instantiations of maliciously OW secure WE.

2.2 The Instantiation from Pairings

We first consider the construction of malicious WE for equivocal vector commitment from pairings. For pairing preliminaries, we adopt an asymmetric bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ of prime order q with pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_q$, we define $[a]_s = g_s^a$.

Our construction is adapted from the Libert-Yung vector commitment [LY10] based on the CBDHE assumption. A key observation is that we can exploit a CBDHE instance to claim the unpredictability of $(\mathsf{m}_0, \mathsf{m}_1)$ and hence

present an efficient malicious WE scheme for equivocal vector commitment as follows:

$$\begin{aligned}
\mathsf{crs}_c &: \overbrace{[\alpha, \alpha^2, \ldots, \alpha^N]_1, [\alpha, \alpha^2, \ldots, \alpha^N, \alpha^{N+2}, \ldots, \alpha^{2N}]_2}^{\text{CBDHE instance}} \\
\mathsf{com} &: \left[r + \textstyle\sum_{j=0}^{N-1} D[j] \cdot \alpha^{j+1}\right]_1 \\
\pi_L &: \left[r\alpha^{N-L} + \textstyle\sum_{j=0, j\neq L}^{N-1} D[j] \cdot \alpha^{N-L+j+1}\right]_2 \\
\mathsf{ct}_b &: [s]_1, \underbrace{\left[sr\alpha^{N-L} + s\sum_{j=0}^{N-1} D[j] \cdot \alpha^{N-L+j+1} - b \cdot s\alpha^{N+1}\right]_T}_{C_b = e(\mathsf{com}/[b\cdot\alpha^{L+1}]_1, [\alpha^{N-L}]_2)^s} \cdot \mathsf{m}_b
\end{aligned} \tag{2}$$

where $r, s \leftarrow \mathbb{Z}_q$, (com, L, b) is the statement and m_b is a random element over $\mathbb{G}_T$. For decryption, the receiver can obtain $\mathsf{m}_{D[L]}$ by recovering

$$C_{D[L]} = e\left([s]_1, \pi_L\right) = \left[sr\alpha^{N-L} + s\sum_{j=0}^{N-1} D[j] \cdot \alpha^{N-L+j+1} - b \cdot s\alpha^{N+1}\right]_T$$

Achieving equivocality. Recall that this property requires: i) simulating a special digest $\widetilde{\mathsf{com}}$ (without knowing D) that is indistinguishable from com; ii) equivocating $\widetilde{\mathsf{com}}$ to $\widetilde{\pi}_{L,b}$ for $b \in \{0,1\}$ that is statistically indistinguishable from π_L when $b = D[L]$. We argue the equivocality of the construction in (2) as follows:

$$\begin{aligned}
\mathsf{com} &: \left[r + \textstyle\sum_{j=0}^{N-1} D[j] \cdot \alpha^{j+1}\right]_1 &&\approx_s \overbrace{[\widetilde{r}]_1}^{\widetilde{\mathsf{com}}} \\
\pi_L &: \left[r\alpha^{N-L} + \textstyle\sum_{j=0, j\neq L}^{N-1} D[j] \cdot \alpha^{N-L+j+1}\right]_2 &&\approx_s \underbrace{[\widetilde{r}\alpha^{N-L} - D[L] \cdot \alpha^{N+1}]_2}_{\widetilde{\pi}_{L,D[L]}}
\end{aligned}$$

where $\widetilde{r} \leftarrow \mathbb{Z}_q$. By setting $\mathsf{td} := \alpha$, both $\widetilde{\pi}_{L,0}$ and $\widetilde{\pi}_{L,1}$ can be properly derived.

Achieving malicious OW security. We see that, if the adversary can recover both m_0 and m_1 from $(\mathsf{ct}_0, \mathsf{ct}_1)$, it must learn (C_0, C_1) and thus compute $C_0/C_1 = [s\alpha^{N+1}]_T$, which is exactly a solution to CBDHE. Therefore, if one can predict $(\mathsf{m}_0, \mathsf{m}_1)$ with non-negligible probability, it can also break the CBDHE assumption.

On the other hand, we can also adapt the vector commitment from [CF13] into an equivocal vector commitment and obtain the other malicious WE scheme based on the standard CBDH assumption in Section 4.5.

2.3 The Instantiation from Lattices

While the pairing-based construction has been provided, it is usually more desirable to have one from lattices for the purpose of achieving post-quantum security. For lattice preliminaries, we let $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, \ldots, 2^{\log q - 1})$ denote the gadget matrix as defined in [MP12] and $\mathbf{G}^{-1}$ denote the binary decomposition operator, i.e., $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{Z}_q^n$.

We present a lattice-based malicious WE for equivocal vector commitment, which is adapted from [DKL+23]. In a nutshell, we use the Merkle-tree-based vector commitment (described in Fig. 1) in this instantiation. The witness encryption for this vector commitment can be viewed as a dual-Regev encryption scheme. As for decryption, it requires a witness π_L that proves a single database bit $D[L]$ is committed at Position L in the Merkle tree.

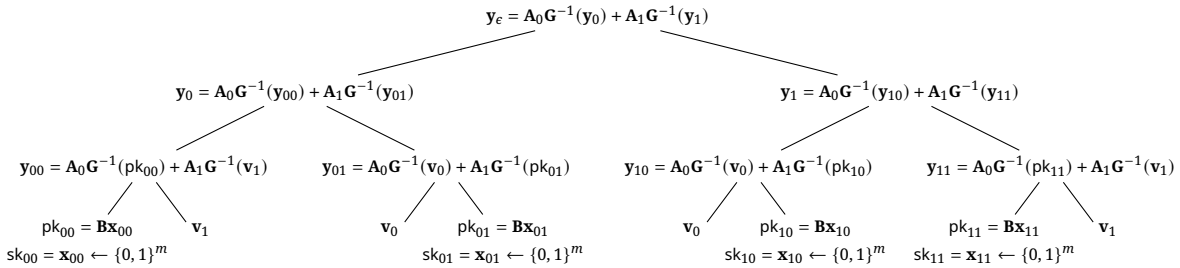

Fig. 1. The Merkle-tree-based commitment $\mathbf{y}_\epsilon$ to a database D. Here, we consider the simple case when $D = 0110$.

More concretely, for a database vector $D \in \{0,1\}^N$, we let $\ell = \log N$ and interpret $L = L[0]\|\cdots\|L[\ell-1]$, then our construction works as follows:

$$\begin{aligned}
&\mathsf{crs}_c : \mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n\times m},\ \mathbf{A}_1 \leftarrow \mathbb{Z}_q^{n\times m},\ \mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m},\ \mathbf{v}_0 \leftarrow \mathbb{Z}_q^{n},\ \mathbf{v}_1 \leftarrow \mathbb{Z}_q^{n}\\
&\mathsf{com} : \mathbf{y}_\epsilon \in \mathbb{Z}_q^n\\
&\quad \pi_L : \left(\mathbf{G}^{-1}(\mathbf{y}_0), \mathbf{G}^{-1}(\mathbf{y}_1), \ldots, \mathbf{G}^{-1}(\mathbf{y}_{L\|0}), \mathbf{G}^{-1}(\mathbf{y}_{L\|1}), \mathbf{x}_L\right) \in \mathbb{Z}_q^{(\ell+1)n}\\
&\quad \mathsf{ct}_b : \underbrace{\mathbf{s}^\top\mathbf{D}} \in \mathbb{Z}_q^{(2\ell+3)m},\ \underbrace{\lfloor -\mathbf{s}_0^\top\mathbf{y}_\epsilon + \mathbf{s}_{\ell+1}^\top\mathbf{v}_{1-b}\rfloor_c} \oplus \mathsf{m}_b
\end{aligned} \tag{3}$$

where the commitment $\mathbf{y}_\epsilon$ is computed as follows:

(i) $\mathsf{sk}_{\mathsf{ind}} := \mathbf{x}_{\mathsf{ind}} \leftarrow \{0,1\}^m$ and $\mathsf{pk}_{\mathsf{ind}} := \mathbf{B}\mathbf{x}_{\mathsf{ind}} \in \mathbb{Z}_q^n$ for all $\mathsf{ind} \in \{0,1\}^\ell$, where $(\mathsf{sk}_{\mathsf{ind}}, \mathsf{pk}_{\mathsf{ind}})$ is essentially a dual-Regev key pair [Reg05];

(ii) $\mathbf{y}_{\mathsf{ind}} = \mathbf{A}_{D[\mathsf{ind}]}\mathbf{G}^{-1}(\mathsf{pk}_{\mathsf{ind}}) + \mathbf{A}_{\overline{D}[\mathsf{ind}]}\mathbf{G}^{-1}(\mathbf{v}_{\overline{D}[\mathsf{ind}]})$ for all $\mathsf{ind} \in \{0,1\}^\ell$, where we treat $\mathbf{y}_{\mathsf{ind}\|D[\mathsf{ind}]} = \mathsf{pk}_{\mathsf{ind}}$ and $\mathbf{y}_{\mathsf{ind}\|\overline{D}[\mathsf{ind}]} = \mathbf{v}_{\overline{D}[\mathsf{ind}]}$;

(iii) $\mathbf{y}_{\mathsf{ind}} = \mathbf{A}_0\mathbf{G}^{-1}(\mathbf{y}_{\mathsf{ind}\|0}) + \mathbf{A}_1\mathbf{G}^{-1}(\mathbf{y}_{\mathsf{ind}\|1})$ for all $\mathsf{ind} \in \{0,1\}^{0<j<\ell}$;

(iv) $\mathbf{y}_\epsilon = \mathbf{A}_0\mathbf{G}^{-1}(\mathbf{y}_0) + \mathbf{A}_1\mathbf{G}^{-1}(\mathbf{y}_1)$.

Besides, $\mathbf{s} = (\mathbf{s}_0, \ldots, \mathbf{s}_{\ell+1}) \leftarrow \mathbb{Z}_q^{(\ell+2)n}$ is the randomness for encryption, and ct_b is indeed a dual-Regev ciphertext under the matrix $\mathbf{D}$ and the vector $\mathbf{d}_b = (-\mathbf{y}_\epsilon, \mathbf{0}, \ldots, \mathbf{0}, \mathbf{v}_{1-b}) \in \mathbb{Z}_q^{(\ell+2)n}$, where

$$\mathbf{D} = \begin{pmatrix}
-\mathbf{A}_0 & -\mathbf{A}_1 & & & & & & & \\
\overline{L}[0]\cdot\mathbf{G} & L[0]\cdot\mathbf{G} & -\mathbf{A}_0 & -\mathbf{A}_1 & & & & & \\
 & & \overline{L}[1]\cdot\mathbf{G} & L[1]\cdot\mathbf{G} & & & & & \\
 & & & & \ddots & \ddots & & & \\
 & & & & & & -\mathbf{A}_0 & -\mathbf{A}_1 & \\
 & & & & & & \overline{L}[\ell-1]\cdot\mathbf{G} & L[\ell-1]\cdot\mathbf{G} & -\mathbf{A}_0 & -\mathbf{A}_1 \\
 & & & & & & & & \mathbf{G} & \mathbf{G} & -\mathbf{B}
\end{pmatrix} \in \mathbb{Z}_q^{(\ell+2)n\times(2\ell+3)m}.$$

We let "$\underbrace{\cdot}$" denote a noised version of the input.

To decrypt ct_b, it can approximately compute

$$\underbrace{\mathbf{s}^\top\mathbf{D}} \cdot \pi_L = \underbrace{-\mathbf{s}_0^\top\mathbf{y}_\epsilon + \mathbf{s}_{\ell+1}^\top\mathbf{v}_{1-b}} + e$$

where e denotes the extra noise disturbance. We can apply a rounding function $\lfloor\cdot\rfloor_c : \mathbb{Z}_q \to \mathbb{Z}_c$ [BPR12] to erase e. In more detail, by carefully choosing the rounding modulus c and the noise bound of e, it holds that $\lfloor\underbrace{\mathbf{s}^\top\mathbf{D}} \cdot \pi_L\rfloor_c = \lfloor\underbrace{-\mathbf{s}_0^\top\mathbf{y}_\epsilon + \mathbf{s}_{\ell+1}^\top\mathbf{v}_{1-b}} + e\rfloor_c = \lfloor\underbrace{\mathbf{s}_0^\top\mathbf{y}_\epsilon + \mathbf{s}_{\ell+1}^\top\mathbf{v}_{1-b}}\rfloor_c$. As a disadvantage, the rounding operation usually maps

$\mathbf{s}_0^\top \mathbf{y}_\epsilon + \mathbf{s}_{\ell+1}^\top \mathbf{v}_{1-b}$ to an element in a smaller space $\mathbb{Z}_c$ and hence increases the guessing advantage of the adversary. This can be resolved via the standard hardness amplification techniques, and we omit more details.

Achieving equivocality. In the reduction proof, the simulator needs to simulate $\widetilde{\mathsf{com}}$ (without knowing D) and equivocate $\widetilde{\mathsf{com}}$ into $\widetilde{\pi}_{L,b}$ for $b \in \{0,1\}$. Our idea is to exploit the good properties of trapdoor preimage sampling from [Ajt96,GPV08,MP12]. In particular, we will modify steps (i)-(iv) in the commitment generation into the following form:

(i) $\mathsf{sk}_{\mathsf{ind}} := \mathbf{x}_{\mathsf{ind}} \leftarrow \mathcal{D}^m$ and $\mathsf{pk}_{\mathsf{ind}} := \mathbf{B}\mathbf{x}_{\mathsf{ind}} \in \mathbb{Z}_q^n$ for all $\mathsf{ind} \in \{0,1\}^\ell$, where $\mathcal{D}$ denotes a discrete Gaussian distribution over $\mathbb{Z}$;

(ii) $\mathbf{y}_{\mathsf{ind}} = \mathbf{A}_{D[\mathsf{ind}]} \overline{\mathbf{G}}^{-1}(\mathsf{pk}_{\mathsf{ind}}) + \mathbf{A}_{\overline{D}[\mathsf{ind}]} \overline{\mathbf{G}}^{-1}(\mathbf{v}_{\overline{D}[\mathsf{ind}]})$ for all $\mathsf{ind} \in \{0,1\}^\ell$;

(iii) $\mathbf{y}_{\mathsf{ind}} = \mathbf{A}_0 \overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}\|0}) + \mathbf{A}_1 \overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}\|1})$ for all $\mathsf{ind} \in \{0,1\}^{0<j<\ell}$;

(iv) $\mathbf{y}_\epsilon = \mathbf{A}_0 \overline{\mathbf{G}}^{-1}(\mathbf{y}_0) + \mathbf{A}_1 \overline{\mathbf{G}}^{-1}(\mathbf{y}_1)$.

where $\overline{\mathbf{G}}^{-1}(\cdot) := \begin{pmatrix} \mathbf{G}^{-1}(\cdot) \\ \mathbf{r} \end{pmatrix}$ and $\mathbf{r}$ is a freshly sampled Gaussian vector for each input. By the trapdoor preimage sampling, $(\mathbf{r}, \mathbf{A}\mathbf{r})$ is statistically indistinguishable from a tuple $(\mathbf{r}', \mathbf{y})$, where $\mathbf{y}$ is uniformly sampled and $\mathbf{r}'$ is sampled using a trapdoor for $\mathbf{A}$. Therefore, for all $\mathsf{ind} \in \{0,1\}^\ell$, the simulator can treat $\mathbf{y}_{\mathsf{ind}}$ (and $\mathsf{pk}_{\mathsf{ind}}$) as a truly random vector, so that it can compute $\mathbf{y}_\epsilon$ just using $\mathbf{y}_{\mathsf{ind}}$, without involving D. On the other hand, we can see that the decryption of the modified WE scheme requires the knowledge of

$$\pi_L : \left(\overline{\mathbf{G}}^{-1}(\mathbf{y}_0), \overline{\mathbf{G}}^{-1}(\mathbf{y}_1), \ldots, \overline{\mathbf{G}}^{-1}(\mathbf{y}_{L\|0}), \overline{\mathbf{G}}^{-1}(\mathbf{y}_{L\|1}), \mathbf{x}_L\right) \in \mathbb{Z}_q^{(\ell+1)n}$$

In the equivocal mode, since the randomness $\mathbf{r}$ and $\mathbf{x}_L$ have not been determined yet, we can use the trapdoors for matrices $(\mathbf{A}_0, \mathbf{A}_1, \mathbf{B})$ to sample them. In particular, we can use the trapdoors to properly sample new Gaussian vectors $\mathbf{r}'$ and $\mathbf{x}'_L$ to form a new proof $\widetilde{\pi}_{L,b}$ for $b \in \{0,1\}$, which is statistically indistinguishable from the above π_L. In this way, we manage to obtain a concrete lattice-based equivocal vector commitment scheme.

Achieving malicious OW security. We see that the matrix $\mathbf{D}$ contains a random matrix component (either $\mathbf{A}_0$, $\mathbf{A}_1$ or $\mathbf{B}$) on each column, and thus the first ciphertext term $\mathbf{s}^\top \mathbf{D}$ is pseudorandom due to the DLWE assumption. The second ciphertext term contains a vector $\mathbf{y}_\epsilon$ that is generated by the malicious adversary, so we cannot apply DLWE here. Nonetheless, since we only require one-way security, this part will be masked by a random message m_b and will also look random. That means, it is safe to replace both ciphertexts ct_0 and ct_1 with random elements. In addition, we can show that any adversary that recovers both random messages, which is required in the malicious OW security, will be able to distinguish $\mathbf{s}_{\ell+1}^\top(\mathbf{v}_0 - \mathbf{v}_1)$ from a random vector, which exactly breaks the DLWE assumption. Then the malicious OW security naturally follows.

3 Preliminaries

Notations. For a finite set S, we write $s \leftarrow S$ to denote that s is picked uniformly from S. Then, we use $|S|$ to denote the size of S. For any integer n, $[n]$ and $[0,n]$ denote sets $\{1,\ldots,n\}$ and $\{0,1,\ldots,n\}$, respectively. Let $\approx_s$ denote two distributions being statistically indistinguishable, and $\approx_c$ denote two distributions being computationally indistinguishable. For any $x \in \{0,1\}^n$, we use $x[w]$ to denote the w-th bit of x and $x[:w]$ to denote the prefixed w bits of x, where the empty string $x[:0]$ is rewritten as ϵ for simplicity. Besides, we let $\overline{x}[w] = 1 - x[w]$. For a vector $\mathbf{v} = (v_1, \ldots, v_m)$, let $\|\mathbf{v}\| := \sqrt{\sum_{i=1}^m v_i^2}$ denote its Euclidean norm.

3.1 Lattice Background

We use $\mathcal{D}_{\mathbb{Z},\sigma}$ to denote the discrete Gaussian distribution over $\mathbb{Z}$ with standard deviation $\sigma > 0$. We review the Decisional Learning with Errors assumption introduced by Regev [Reg05].

Lemma 1 (Gaussian Tail Bound [MP12]). *Given $\lambda, n, m, q \in \mathbb{N}$ such that $m \geq 2n \log q$. For (almost) all matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and all $\sigma \geq \log m$, we have*

$$\Pr[|x| \geq B] \leq 2^{-\lambda}$$

where $x \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}$ and $B = \sqrt{\lambda}\sigma$.

Assumption 1 (DLWE assumption [Reg05]) *Given $n, m, q, \sigma \in \mathbb{N}$, we say the $\mathsf{DLWE}_{n,m,q,\sigma}$ assumption holds if*

$$(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top) \approx_c (\mathbf{A}, \mathbf{u}^\top)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$ and $\mathbf{u} \leftarrow \mathbb{Z}_q^m$.

Here, to claim the hardness of DLWE assumption, we assume $n = O(\lambda), m \geq 2n \log q, \sigma = O(\lambda)$ and q/B is polynomial in λ. In this work, we consider a matrix variant of the DLWE assumption.

Assumption 2 (Matrix DLWE assumption [Mic18]) *Given $n, m, k, q, \sigma \in \mathbb{N}$, we say the matrix $\mathsf{DLWE}_{n,m,k,q,\sigma}$ assumption holds if*

$$(\mathbf{A}, \mathbf{S}^\top \mathbf{A} + \mathbf{E}^\top) \approx_c (\mathbf{A}, \mathbf{U}^\top)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{S} \leftarrow \mathbb{Z}_q^{n \times k}, \mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m \times k}$ and $\mathbf{U} \leftarrow \mathbb{Z}_q^{m \times k}$.

If $k = \mathsf{poly}(\lambda)$, we can reduce the hardness of the matrix DLWE assumption to that of the DLWE assumption via a standard hybrid argument.

Assumption 3 (SIS assumption [GPV08]) *Given $n, m, q, \delta \in \mathbb{N}$, we say the $\mathsf{SIS}_{n,m,q,\delta}$ assumption holds if for all efficient adversaries $\mathcal{A}$,*

$$\Pr[\mathcal{A}(\mathbf{A}) = \mathbf{z}] = \mathsf{negl}(n)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \mod q$ and $\|\mathbf{z}\| \leq \delta$.

In this work, we also consider an inhomogeneous variant of the SIS assumption.

Assumption 4 (Inhomogeneous SIS assumption [GPV08]) *Given $n, m, q, \delta \in \mathbb{N}$, we say the inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ assumption holds if for all efficient adversaries $\mathcal{A}$,*

$$\Pr[\mathcal{A}(\mathbf{A}, \mathbf{v}) = \mathbf{z}] = \mathsf{negl}(n)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{v} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{v} \mod q$ and $\|\mathbf{z}\| \leq \delta$.

Here, to claim the hardness of (inhomogeneous) SIS assumption, we assume $q \geq \delta \cdot \widetilde{O}(\sqrt{n})$ and $m \geq 2n \log q$.

Gadget Matrix. For matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{k \times \ell}$, we write $\mathbf{A} \otimes \mathbf{B} \in \mathbb{Z}_q^{nk \times m\ell}$ to denote their tensor (Kronecker) product. For any $n, q \in \mathbb{N}$, let $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^\top \in \mathbb{Z}_q^{n \times m'}$ be the gadget matrix [MP12] where $\mathbf{I}_n$ is the identity matrix of dimension n, $\mathbf{g}^\top = (1, 2, \ldots, 2^{\log q - 1})$ and $m' = n \log q$. We write $\mathbf{G}^{-1} : \mathbb{Z}_q^n \to \{0,1\}^{m'}$ to denote the operator that expands each component of the input into its binary representation (i.e., $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{Z}_q^n$). We can define an extended gadget matrix $\overline{\mathbf{G}} = (\mathbf{G} \mid \mathbf{0}^{n \times m'})$ along with $\overline{\mathbf{G}}^{-1}(\mathbf{x}) = \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{x}) \\ \mathbf{r} \end{pmatrix}$ where $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m'}$. Still, it holds that $\overline{\mathbf{G}} \cdot \overline{\mathbf{G}}^{-1}(\mathbf{x}) = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{Z}_q^n$. Besides, for a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a target $\mathbf{y} \in \mathbb{Z}_q^n$ in the image of $\mathbf{A}$, we write $\mathbf{A}_\sigma^{-1}(\mathbf{y})$ to denote the random variable $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m$ conditioned on $\mathbf{A}\mathbf{x} = \mathbf{y}$.

Lemma 2 (Gaussian Preimages [GPV08]). *Given $n, m, q \in \mathbb{N}$ such that $m \geq 2n \log q$. For (almost) all matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and all $\sigma \geq \log m$, we have*

$$(\mathbf{x}, \mathbf{Ax}) \approx_s (\mathbf{z}, \mathbf{y})$$

where $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^m, \mathbf{z} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{y})$ and $\mathbf{y} \leftarrow \mathbb{Z}_q^n$.

Gadget Trapdoor. We recall the gadget trapdoor from [Ajt96,GPV08,MP12]: given $n, m, q \in \mathbb{N}$ such that $m \geq 2n \log q$, there exists the following algorithms:

- $\mathsf{TGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T})$. It takes as input the lattice parameters n, m, q, outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ along with a trapdoor $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$.
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{y}, \sigma) \to \mathbf{x}$. It takes as input the matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a trapdoor $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$, a target vector $\mathbf{y} \in \mathbb{Z}_q^n$ and a width parameter $\sigma > 0$, outputs a preimage $\mathbf{x} \in \mathbb{Z}_q^m$.

It requires that (i) the distribution of $\mathbf{A}$ is statically close to the uniform distribution over $\mathbb{Z}_q^{n \times m}$, where we can assume $\|\mathbf{T}\| = 1$; (ii) $\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{y}, \sigma)$ satisfies $\mathbf{Ax} = \mathbf{y}$; (iii) it holds that $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{y}, \sigma) \approx_s \mathbf{A}_\sigma^{-1}(\mathbf{y})$ when $\sigma \geq m\|\mathbf{T}\| \log n$.

Rounding Function. For any $c, q \in \mathbb{N}$ such that $q = c \cdot p$ is an integer modulus, let $\lfloor \cdot \rfloor_c : \mathbb{Z}_q \to \mathbb{Z}_c$ be the rounding function [BPR12] where $\lfloor x \rfloor_c = \lfloor \widetilde{x} \cdot c/q \rfloor \mod c$ for all $x \in \mathbb{Z}_q$, and $\widetilde{x} \in \mathbb{Z}$ is an arbitrary residue-class representative of x.

Lemma 3 ([BPR12]). *Given $c, q \in \mathbb{N}$ where $q = c \cdot p$ is an integer modulus, let $x \leftarrow \mathbb{Z}_q$ be distributed uniformly at random. Then for all $v \in \{-B, \ldots, B\}$, it holds that*

$$\Pr\left[\lfloor x \rfloor_c = \lfloor x + v \rfloor_c\right] = 1 - (2B + 1) \cdot c/q$$

3.2 Pairing Background

A generator $\mathcal{G}$ takes as input a security parameter 1^λ and outputs a description $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, where q is a prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order q, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear map. Group operations in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and bilinear map e are computable in deterministic polynomial time in λ. Let $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $g_T = e(g_1, g_2) \in \mathbb{G}_T$ be the respective generators, we employ *implicit representation* of group elements: for a value $x \in \mathbb{Z}_p$, we define $[x]_s = g_s^x, \forall s \in \{1, 2, T\}$, where exponentiation is carried out component-wise. Given $[x]_1$ and $[y]_2$, we let $e([x]_1, [y]_2) = [x \cdot y]_T$. We review the computational (Bilinear) Diffie-Hellman (Exponent) assumptions introduced by Boneh et al. [BF01,BGW05].

Assumption 5 (CDHE assumption [BGW05] over $\mathbb{G}_2$) *Given $\lambda, n \in \mathbb{N}$, we say the CDHE_n assumption over $\mathbb{G}_2$ holds if for all efficient adversaries $\mathcal{A}$,*

$$\Pr\left[\mathcal{A}\left(\mathbb{G}, \{[\alpha^j]_1\}_{j \in [n]}, \{[\alpha^j]_2\}_{j \in [2n] \setminus \{n+1\}}\right) = [\alpha^{n+1}]_2\right] = \mathsf{negl}(\lambda)$$

where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and $\alpha \leftarrow \mathbb{Z}_q$.

Assumption 6 (CBDHE assumption [BGW05]) *Given $\lambda, n \in \mathbb{N}$, we say the CBDHE_n assumption holds if for all efficient adversaries $\mathcal{A}$,*

$$\Pr\left[\mathcal{A}\left(\mathbb{G}, \{[\alpha^j]_1\}_{j \in [n]}, \{[\alpha^j]_2\}_{j \in [2n] \setminus \{n+1\}}, [s]_1\right) = [s\alpha^{n+1}]_T\right] = \mathsf{negl}(\lambda)$$

where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and $\alpha, s \leftarrow \mathbb{Z}_q$.

Assumption 7 (CDH assumption [BF01] over $\mathbb{G}_2$) *Given $\lambda \in \mathbb{N}$, we say the* CDH *assumption over $\mathbb{G}_2$ holds if for all efficient adversaries $\mathcal{A}$,*

$$\Pr[\mathcal{A}(\mathbb{G}, [v]_1, [w]_1, [v]_2, [w]_2) = [vw]_2] = \mathsf{negl}(\lambda)$$

where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and $v, w \leftarrow Z_q$.

Assumption 8 (CBDH assumption [BF01]) *Given $\lambda \in \mathbb{N}$, we say the* CBDH *assumption holds if for all efficient adversaries $\mathcal{A}$,*

$$\Pr[\mathcal{A}(\mathbb{G}, [s]_1, [v]_1, [w]_1, [v]_2, [w]_2) = [svw]_T] = \mathsf{negl}(\lambda)$$

where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$ and $s, v, w \leftarrow Z_q$.

In this work, we consider a square variant of the CDH and CBDH assumptions. For square CDH assumption, the probability that the adversary recovers $[v^2]_2$ from $(\mathbb{G}, [v]_1, [v]_2)$ is negligible. For square CBDH assumption, the probability that the adversary recovers $[sv^2]_T$ from $(\mathbb{G}, [s]_1, [v]_1, [v]_2)$ is negligible.

4 Malicious Witness Encryption for Equivocal Vector Commitment

In this section, we give the formal definition and constructions of malicious witness encryption for equivocal vector commitment. In a nutshell, the new notion additionally needs the vector commitment to be equivocal and the witness encryption to have malicious security. We give the definitions of these two primitives in Section 4.1 and Section 4.2, respectively. Then in Section 4.3, we show how to upgrade a WE for equivocal vector commitment with a weaker version of malicious security to the one with fully malicious security. We then provide instantiations from different assumptions from Section 4.4 to Section 4.6.

4.1 Equivocal Vector Commitment

First, we present the definition of equivocal vector commitment (adapted from [WW23]), which consists of seven algorithms as follows:

- $\mathsf{Setup}(1^\lambda, N) \to \mathsf{crs}$. It takes as input the security parameter 1^λ and the vector length N, then outputs a common reference string crs.
- $\mathsf{Com}(\mathsf{crs}, D) \to (\mathsf{com}, \mathsf{aux})$. It takes as input a commit reference string crs, a binary vector $D \in \{0,1\}^N$, then outputs a vector commitment com and auxiliary information aux.
- $\mathsf{Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L) \to \pi_L$. It takes as input a commit reference string crs, a vector commitment com, auxiliary information aux and a location $L \in [0, N-1]$, then outputs an opening proof π_L.
- $\mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, b, \pi_L) \to 0/1$. It takes as input a commit reference string crs, a vector commitment com, a location $L \in [0, N-1]$, a bit b, and an opening proof π_L, then outputs a bit indicating whether π_L is a valid opening of com to b at location L.
- $\mathsf{TSetup}(1^\lambda, N) \to (\widetilde{\mathsf{crs}}, \mathsf{td})$. It takes as input the security parameter 1^λ and the vector length N, then outputs a simulated common reference string $\widetilde{\mathsf{crs}}$ and trapdoor information td.
- $\mathsf{TCom}(\widetilde{\mathsf{crs}}, \mathsf{td}) \to (\widetilde{\mathsf{com}}, \mathsf{st})$. It takes as input a simulated common reference string $\widetilde{\mathsf{crs}}$ and a trapdoor td, then outputs a simulated commitment $\widetilde{\mathsf{com}}$ and an internal state st.
- $\mathsf{Equiv}(\widetilde{\mathsf{crs}}, \mathsf{td}, \mathsf{st}, L, b) \to \widetilde{\pi}_{L,b}$. It takes as input a simulated common reference string $\widetilde{\mathsf{crs}}$, a trapdoor td, an internal state st, a location $L \in [0, N-1]$ and a bit b, then outputs an equivocated opening proof $\widetilde{\pi}_{L,b}$.

Correctness. An equivocal vector commitment is perfectly correct if for all $\lambda, N \in \mathbb{N}$ and all binary vectors $D = D[0]\| \cdots \|D[N-1] \in \{0,1\}^N$, it holds that

$$\Pr\left[\mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, D[L], \pi_L) = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{com}, \mathsf{aux}) \leftarrow \mathsf{Com}(\mathsf{crs}, D) \\ \pi_L \leftarrow \mathsf{Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L) \end{array}\right] = 1.$$

Efficiency. For all $\lambda \in \mathbb{N}$ and all $D \in \{0,1\}^N$, it holds that $|\mathsf{com}| = \mathsf{poly}(\lambda, \log N)$. Besides, the runtime of Ver is bounded by $\mathsf{poly}(\lambda, \log N)$.

Equivocality. For all $\lambda, N \in \mathbb{N}$, all $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N)$, all $(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{TSetup}(1^\lambda, N)$, all vectors $D = D[0]\| \cdots \|D[N-1]\{0,1\}^N$ and all locations $L \in [0, N-1]$, it requires

$$\left\{\mathsf{crs}, \mathbf{x}, \mathsf{com}, \{D[L], \pi_L\}_{L\in[0,N-1]}\right\} \approx_s \left\{\widetilde{\mathsf{crs}}, \mathbf{x}, \widetilde{\mathsf{com}}, \{D[L], \widetilde{\pi}_{L,D[L]}\}_{L\in[0,N-1]}\right\}$$

where $(\widetilde{\mathsf{com}}, \mathsf{st}) \leftarrow \mathsf{TCom}(\widetilde{\mathsf{crs}}, \mathsf{td})$, and $\widetilde{\pi}_{L,D[L]} \leftarrow \mathsf{Equiv}(\widetilde{\mathsf{crs}}, \mathsf{td}, \mathsf{st}, L, D[L])$.

Location Binding. For all efficient adversaries $\mathcal{A}$, this property requires that

$$\Pr\left[\begin{array}{l} \mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, 0, \pi'_{L,0}) = 1 \\ \wedge\, \mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, 1, \pi'_{L,1}) = 1 \end{array} \;\middle|\; \begin{array}{l} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda, N) \\ \left(\mathsf{com}, L, \pi'_{L,0}, \pi'_{L,1}\right) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array}\right] = \mathsf{negl}(\lambda)$$

4.2 Malicious Witness Encryption

Next, we define the malicious witness encryption for a given language of equivocal vector commitment, which consists of two algorithms as follows:

- $\mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{m}; r) \rightarrow \mathsf{ct}$. It takes as input a common reference string crs (of equivocal vector commitment), a commitment com, a location $L \in [0, N-1]$, a bit $b \in \{0,1\}$, where it defines the statement $s = (\mathsf{com}, L, b)$, and a message $\mathsf{m} \in M$, then outputs a ciphertext ct. Here, $r \in R$ is the randomness.
- $\mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L) \rightarrow \mathsf{m}$. It takes as input a common reference string crs, a ciphertext ct, a statement $s = (\mathsf{com}, L, b)$ and an opening proof π_L, where it defines the witness $w = \pi_L$, then outputs a message $\mathsf{m} \in M$.

Correctness. A malicious witness encryption for equivocal vector commitment is δ-correct if for all $\lambda, N \in \mathbb{N}$, all $D = D[0]\| \cdots \|D[N-1] \in \{0,1\}^N$, all memory locations $L \in [0, N-1]$ and all messages $\mathsf{m} \in M$, it holds that

$$\Pr\left[\mathsf{m} = \mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, D[L]), \pi_L) \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{com}, \mathsf{aux}) \leftarrow \mathsf{Com}(\mathsf{crs}, D) \\ \pi_L \leftarrow \mathsf{Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, x[L]), \mathsf{m}) \end{array}\right] = 1 - \delta.$$

Besides, it is also called perfectly correct if $\delta = 0$.

Efficiency. The overhead of Enc depends on the overhead of verifying whether $s = (\mathsf{com}, L, b)$ is a false statement, i.e., checking $\mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, b, \cdot) \stackrel{?}{=} 0$. Based on the efficiency requirements of equivocal vector commitment, we naturally have $|\mathsf{ct}| = \mathsf{poly}(\lambda, \log N)$ and the runtime of Enc is bounded by $\mathsf{poly}(\lambda, \log N)$.

Malicious OW Security. For all $\lambda, N \in \mathbb{N}$, and all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the malicious one-way (OW) security requires

$$\Pr\left[(\mathsf{m}_0, \mathsf{m}_1) = (\mathsf{m}'_0, \mathsf{m}'_1) \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{com}, L, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{crs}); \\ \mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 0), \mathsf{m}_0; r) \\ \mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 1), \mathsf{m}_1; r) \\ (\mathsf{m}'_0, \mathsf{m}'_1) \leftarrow \mathcal{A}_2(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{st}) \end{array}\right] = \mathsf{negl}(\lambda)$$

where $\mathsf{m}_0, \mathsf{m}_1 \leftarrow M$ and $r \leftarrow R$.

Malicious IND Security. For all $\lambda, N \in \mathbb{N}$, and all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the malicious IND security defines the following advantage

$$\mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A},\beta}(\lambda) = \left| \Pr\left[\alpha = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{com}, L, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{crs}) \\ \mathsf{ct}_\beta \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, \beta), \mathsf{m}_\beta; r) \\ \mathsf{ct}_{1-\beta} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 1-\beta), \mathsf{m}_{1-\beta}; r) \\ \alpha \leftarrow \mathcal{A}_2(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{st}) \end{array}\right] - \Pr\left[\alpha = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{com}, L, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{crs}) \\ \mathsf{ct}_\beta \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, \beta), \widetilde{\mathsf{m}}; r) \\ \mathsf{ct}_{1-\beta} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 1-\beta), \mathsf{m}_{1-\beta}; r) \\ \alpha \leftarrow \mathcal{A}_2(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{st}) \end{array}\right] \right|$$

where $\beta \in \{0,1\}$, $\widetilde{\mathsf{m}} \leftarrow M$ and $r \leftarrow R$. This security notion requires

$$\Pr\left[\mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A},0}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)} \wedge \mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A},1}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)}\right] = \mathsf{negl}(\lambda).$$

4.3 From Malicious OW Security to Malicious IND Security

Construction in the Standard Model. Assume a witness encryption scheme $\mathsf{WE} = (\mathsf{Enc}, \mathsf{Dec})$ with malicious OW security for an equivocal vector commitment $\mathsf{EVC} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{Open}, \mathsf{Ver}, \mathsf{TSetup}, \mathsf{TCom}, \mathsf{Equiv})$, we present a generic construction with malicious IND security as follows:

- $\mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{m})$: Suppose $\mathsf{m} \in \{0,1\}$, then it proceeds as follows:
  - Sample randomness $r = (\{\mathsf{k}_{i,\beta}, r_i\}_{i\in[\lambda],\beta\in\{0,1\}}, t_0, t_1)$, where $\mathsf{k}_{i,\beta} \leftarrow \{0,1\}$ and $t_0, t_1 \leftarrow \{0,1\}^\lambda$.
  - For all $i \in [\lambda]$, generate $\mathsf{ct}_i \leftarrow \mathsf{WE.Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{k}_{i,b}; r_i)$. Let $\mathsf{k}_b = \mathsf{k}_{1,b} \| \cdots \| \mathsf{k}_{\lambda,b}$.
  - Compute $C = \mathsf{m} \oplus \mathsf{GL}(\mathsf{k}_b, t_b)$, where the Goldreich-Levin hardcore predicate $\mathsf{GL}(\mathsf{k}_b, t_b) := \left(\bigoplus_{j\in[\lambda]} \mathsf{k}_{j,b} \cdot t_b[j]\right)$.

  Output $\mathsf{ct} = (\{\mathsf{ct}_i\}_{i\in[\lambda]}, C, t_b)$.
- $\mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L)$: For all $i \in [\lambda]$, it recovers $\mathsf{k}_{i,b} \leftarrow \mathsf{WE.Dec}(\mathsf{crs}, \mathsf{ct}_i, (\mathsf{com}, L, b), \pi_L)$. Let $\mathsf{k}_b = \mathsf{k}_{1,b} \| \cdots \| \mathsf{k}_{\lambda,b}$, output $\mathsf{m} = C \oplus \mathsf{GL}(\mathsf{k}_b, t_b)$.

Its correctness naturally follows the underlying WE scheme. As for efficiency, observe that this construction only enlarges the $O(\lambda)\times$ encryption and decryption costs of the underlying WE scheme, which still satisfy the efficiency requirement.

Theorem 1. *Assuming* WE *satisfies the malicious OW security, then the above construction achieves the malicious IND security in the standard model.*

Proof. The proof strategy is analogous to [DGH+20]. First, we can rely on the standard hardness amplification technique [CHS05] to claim the unpredictability: it must be infeasible for the adversary to predict m_β for some $\beta \in \{0,1\}$. In particular, for all $\lambda, N \in \mathbb{N}$, and all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we can define

$$\mathsf{Adv}^{\mathsf{UNP}}_{\mathcal{A},\beta}(\lambda) = \Pr\left[\bigwedge_{i\in[\lambda]} (\mathsf{m}'_{i,\beta} = \mathsf{m}_{i,\beta}) \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{EVC.Setup}(1^\lambda, N) \\ (\mathsf{com}, L, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{crs}) \\ \mathsf{ct}_{i,0} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 0), \mathsf{m}_{i,0}; r_i) \\ \mathsf{ct}_{i,1} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 1), \mathsf{m}_{i,1}; r_i) \\ \{\mathsf{m}'_{i,\beta}\}_{i\in[\lambda]} \leftarrow \mathcal{A}_2(\{\mathsf{ct}_{i,0}, \mathsf{ct}_{i,1}\}_{i\in[\lambda]}, \beta, \mathsf{st}) \end{array}\right]$$

where $\beta \in \{0,1\}$, $\mathsf{m}_{i,0}, \mathsf{m}_{i,1} \leftarrow M$ and $r_i \leftarrow R$. It holds that

$$\Pr\left[\mathsf{Adv}^{\mathsf{UNP}}_{\mathcal{A},0}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)} \wedge \mathsf{Adv}^{\mathsf{UNP}}_{\mathcal{A},1}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)}\right] = \mathsf{negl}(\lambda).$$

Lemma 4. *Assuming witness encryption satisfies the malicious OW security, our construction presented in Section 4.3 satisfies the unpredictability as defined above.*

Proof. if there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such as

$$\Pr\left[\mathsf{Adv}^{\mathsf{UNP}}_{\mathcal{A},0}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)} \wedge \mathsf{Adv}^{\mathsf{UNP}}_{\mathcal{A},1}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)}\right] = \frac{1}{\mathsf{poly}(\lambda)}.$$

Equivalently, for all $\beta \in \{0,1\}$, it will have

$$\Pr[\mathcal{A}_2(\{\mathsf{ct}_{i,0}, \mathsf{ct}_{i,1}\}_{i\in[\lambda]}, \beta, \mathsf{st}) = \{\mathsf{m}_{i,\beta}\}_{i\in[\lambda]}] > \frac{1}{\mathsf{poly}(\lambda)}$$

where $\mathsf{ct}_{i,0} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 0), \mathsf{m}_{i,0}; r_i)$, $\mathsf{ct}_{i,1} \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, 1), \mathsf{m}_{i,1}; r_i)$, $\mathsf{m}_{i,0}, \mathsf{m}_{i,1} \leftarrow M$ and $r_i \leftarrow R$. Then by Lemma 5, we will have

$$\Pr[\mathsf{Amp}^{\mathcal{A}_2(\cdot,\beta,\mathsf{st}),\mathsf{Enc}}(1^\lambda, \mathsf{ct}_0, \mathsf{ct}_1) = \mathsf{m}_\beta] \geq c$$

where $\beta \in \{0,1\}$, $\mathsf{ct}_\beta \leftarrow \mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, \beta), \mathsf{m}_\beta; r)$, $\mathsf{m}_\beta \leftarrow M$ and $r \leftarrow R$. Let $c = \frac{3}{4}$, then we can construct an algorithm $\mathcal{B}$ against the malicious OW security, which takes as input $(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{st})$, then outputs $\mathsf{Amp}^{\mathcal{A}_2(\cdot,\beta,\mathsf{st}),\mathsf{Enc}}$ for all $\beta \in \{0,1\}$. Therefore, we can see that $\Pr[\mathcal{B}(\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{st}) = (\mathsf{m}_0, \mathsf{m}_1)] \geq 1 - \sum_{\beta\in\{0,1\}} \Pr[\mathsf{Amp}^{\mathcal{A}_2(\cdot,\beta,\mathsf{st}),\mathsf{Enc}}(1^\lambda, \mathsf{ct}_0, \mathsf{ct}_1) \neq \mathsf{m}_\beta] \geq \frac{1}{2}$. Therefore, the lemma follows readily. □

Lemma 5 (Hardness Amplification [CHS05]). *Let Y be some distribution over pairs of puzzles and solutions* (p, s) *and let* Solve *be an algorithm such that*

$$\Pr[\mathsf{Solve}(\{\mathsf{p}_i\}_{i\in[\lambda]}) = \{\mathsf{s}_i\}_{i\in[\lambda]}] \geq \frac{1}{\mathsf{poly}(\lambda)}$$

where $(\mathsf{p}_i, \mathsf{s}_i) \leftarrow Y$ *for all* $i \in [\lambda]$. *Then for all constant* $c > 0$ *there exists an algorithm* Amp *such that*

$$\Pr[\mathsf{Amp}^{\mathsf{Solve},Y}(1^\lambda, \mathsf{p}) = \mathsf{s}] \geq c$$

where $(\mathsf{p}, \mathsf{s}) \leftarrow Y$.

For our construction in Section 4.3, by Lemma 4, we can see that k_β for some $\beta \in \{0,1\}$ is unpredictable given $(\mathsf{ct}_{i,0}, \mathsf{ct}_{i,1})$, so we can make a standard search-to-decision reduction via Goldreich-Levin hardcore predicate to claim the pseudorandomness of k_β via Lemma 6.

Lemma 6 (Goldreich-Levin [GL89,HLR07,HWW25]). *Let Y be some distribution over pairs of values* (p, s) *where* $\mathsf{s} \in \{0,1\}$. *For* $(\mathsf{p}_1, \mathsf{s}_1), \ldots, (\mathsf{p}_\lambda, \mathsf{s}_\lambda) \leftarrow Y$, *let* $\mathsf{S} = \mathsf{p}_1 \| \cdots \| \mathsf{p}_\lambda$ *and* $\mathsf{S} = \mathsf{s}_1 \| \cdots \| \mathsf{s}_\lambda$, *and let* $\mathsf{GL}(\mathsf{S}, t) := \left(\bigoplus_{j\in[\lambda]} \mathsf{s}_j \cdot t[j]\right)$ *be the Goldreich-Levin hardcore predicate (as specified in Section 4.3). For all efficient adversaries $\mathcal{A}$, if* S *is unpredictable even given* P, *then we will have*

$$\Pr[\mathcal{A}(1^\lambda, t, \mathsf{P}, \mathsf{GL}(\mathsf{P}, t)) = 1] - \Pr[\mathcal{A}(1^\lambda, t, \mathsf{P}, b) = 1] = \mathsf{negl}(\lambda)$$

where $t \leftarrow \{0,1\}^\lambda$ *and* $b \leftarrow \{0,1\}$.

In our construction, since the message m_β is encrypted under $\mathsf{GL}(\mathsf{k}_\beta, t_\beta)$ via one-time padding, we can naturally exploit the pseudorandomness to successfully blind m_β, thus proving the malicious IND security. Finally, we conclude the whole proof of malicious IND security. □

Construction in the Random Model. If one wants to remove the additional $O(\lambda)$ costs in the standard construction, we can turn to rely on the random oracle. Concretely, let $\mathsf{H} : \{0,1\}^* \to \{0,1\}$ be a hash function that can be modeled as a random oracle, the construction now works as follows:

- $\mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{m})$: Suppose $\mathsf{m} \in \{0,1\}$, then it proceeds as follows:
  - Sample randomness $\mathsf{k}_0, \mathsf{k}_1, r \leftarrow \{0,1\}^\lambda$.
  - Generate $\mathsf{ct} \leftarrow \mathsf{WE.Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{k}_b; r)$.
  - Compute $C = \mathsf{m} \oplus \mathsf{H}(\mathsf{k}_b)$.

  Output $\mathsf{ct} = (\mathsf{ct}, C)$.
- $\mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L)$: It recovers $\mathsf{k}_b \leftarrow \mathsf{WE.Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L)$. Output $\mathsf{m} = C \oplus \mathsf{H}(\mathsf{k}_b)$.

Theorem 2. *Assuming* WE *satisfies the malicious OW security, then the above construction achieves the malicious IND security in the random oracle model.*

Proof. If the adversary $\mathcal{A}$ can break the malicious IND security, we can build an algorithm $\mathcal{B}$ to break the malicious OW security.

Initially, $\mathcal{B}$ invokes the challenger in the malicious OW experiment and receives a common reference string crs. Then, $\mathcal{B}$ forwards crs to $\mathcal{A}$ who subsequently specifies a commitment com, a location L, a pair of messages $(\mathsf{m}_0, \mathsf{m}_1)$, and state information st. In the meantime, $\mathcal{B}$ initializes a random oracle $\mathsf{H} : \{0,1\}^* \to \{0,1\}$, and for the i-th query k_i made by $\mathcal{A}$, $\mathcal{B}$ replies with a randomly sampled bit b_i. $\mathcal{B}$ forwards $(\mathsf{com}, L, \mathsf{st})$ to the challenger and obtains ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$. Assuming $\mathcal{A}$ makes totally Q random oracle queries, $\mathcal{B}$ picks $i, j \leftarrow [Q]$ and computes $C_0 = \mathsf{m}_0 \oplus b_i$ and $C_1 = \mathsf{m}_1 \oplus b_j$. Return $\mathsf{ct}'_0 = (\mathsf{ct}_0, C_0)$ and $\mathsf{ct}'_1 = (\mathsf{ct}_1, C_1)$ to $\mathcal{A}$.

Since $\mathcal{A}$ can always decrypt ct_β for some $\beta \in \{0,1\}$, we assume that ct_β encrypts k_i that has been queried to H. We see that if there exists some k' such that $\mathsf{ct}_{1-\beta}$ encrypts k' and it has never been queried, then at least one of (C_0, C_1) is randomly sampled, meaning that $\mathcal{A}$ cannot break the malicious IND security with non-negligible probability. Thus, by contradiction, we can assume that $(\mathsf{ct}_\beta, \mathsf{ct}_{1-\beta})$ encrypt queried keys $(\mathsf{k}_i, \mathsf{k}_j)$. If $\mathcal{A}$ can break the malicious IND security with non-negligible probability ϵ, then $\mathcal{B}$ just outputs $(\mathsf{k}_i, \mathsf{k}_j)$ to break the malicious OW security with non-negligible probability ϵ/Q^2. This completes the proof of Theorem 2. □

4.4 The Construction from CBDHE

First, we show the equivocal vector commitment adapted from [LY10] as follows:

- $\mathsf{Setup}(1^\lambda, N)$: Generate $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $\alpha \leftarrow \mathbb{Z}_q$ and output

$$\mathsf{crs} = (\mathbb{G}, \{[\alpha^j]_1\}_{j\in[N]}, \{[\alpha^j]_2\}_{j\in[2N]\setminus\{N+1\}}).$$

- $\mathsf{Com}(\mathsf{crs}, D)$: Parse $D = D[0]\|\cdots\|D[N-1] \in \{0,1\}^N$. Sample $r \leftarrow \mathbb{Z}_q$ and output $\mathsf{aux} = (D, r)$ and

$$\mathsf{com} = \left[\overbrace{r + \sum_{j=0}^{N-1} D[j]\cdot\alpha^{j+1}}^{d}\right]_1.$$

- $\mathsf{Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L)$: Output $\pi_L = \left[r\alpha^{N-L} + \sum_{j=0, j\neq L}^{N-1} D[j]\cdot\alpha^{N-L+j+1}\right]_2$.
- $\mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, b, \pi_L)$: Parse $\mathsf{com} = [d]_1$ and check whether

$$e([d - b\cdot\alpha^{L+1}]_1, [\alpha^{N-L}]_2) \stackrel{?}{=} e([1]_1, \pi_L).$$

If pass, output 1; otherwise, output 0.

– $\mathsf{TSetup}(1^\lambda, N)$: It generates $\widetilde{\mathsf{crs}}$ as the crs in Setup and output $\mathsf{td} = \alpha$.
– $\mathsf{TCom}(\widetilde{\mathsf{crs}}, \mathsf{td})$: Sample $\widetilde{r} \leftarrow \mathbb{Z}_q$ and output $\widetilde{\mathsf{com}} = [\widetilde{r}]_1$ and $\mathsf{st} = \widetilde{r}$.
– $\mathsf{Equiv}(\widetilde{\mathsf{crs}}, \mathsf{td}, \mathsf{st}, L, b)$: Output $\widetilde{\pi}_{L,b} = \left[(\widetilde{r} - b \cdot \alpha^{L+1})\alpha^{N-L}\right]_2$.

Next, we present a concrete malicious WE for above commitment scheme as follows:

– $\mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{m})$: Parse $\mathsf{com} = [d]_1$ and $L \in [0, N-1]$. Sample $s \leftarrow \mathbb{Z}_q$ and output

$$\mathsf{ct} = \left(\underbrace{[\ s\ ]_1}_{c}, \underbrace{\left[sd\alpha^{N-L} - b \cdot s\alpha^{N+1}\right]_T \cdot \mathsf{m}}_{C} \right)$$

– $\mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L)$: Parse $\mathsf{ct} = ([c]_1, C)$. Output $\mathsf{m} = C/e([c]_1, \pi_L)$.

Correctness and Efficiency. For the correctness of commitment scheme, if $b = D[L]$, we have

$$\begin{aligned} e([1]_1, \pi_L) \cdot e([b \cdot \alpha^{L+1}]_1, [\alpha^{N-L}]_2) &= \left[r\alpha^{N-L} + \sum_{j=0, j\neq L}^{N-1} D[j] \cdot \alpha^{N-L+j+1} + D[L] \cdot \alpha^{N+1} \right]_T \\ &= \left[r\alpha^{N-L} + \sum_{j=0}^{N-1} D[j] \cdot \alpha^{N-L+j+1} \right]_T \\ &= \left[d\alpha^{N-L} \right]_T, \end{aligned}$$

so the verification will pass and the commitment scheme is perfectly correct.

For the correctness of encryption scheme, if $b = D[L]$, then we have $e([c_0]_1, \pi_L) = [sr\alpha^{N-L} + s\sum_{j=0}^{N-1} D[j] \cdot \alpha^{N-L+j+1} - D[L] \cdot s\alpha^{N+1}]_T = [sd\alpha^{N-L} - D[L] \cdot s\alpha^{N+1}]_T = C/\mathsf{m}$. Therefore, $C/e([c_0]_1, \pi_L)$ exactly equals m, so the encryption scheme is perfectly correct.

For efficiency, assuming each group element of size $O(\lambda)$, we can see that $|\mathsf{com}| = O(\lambda), |\mathsf{ct}| = O(\lambda)$ and $(\mathsf{Ver}, \mathsf{Enc})$ only involve a constant number of exponent and pairing operations, which is independent of N.

Equivocality. Apparently, if we assume $r = \widetilde{r} - \sum_{j=0}^{N-1} D[j] \cdot \alpha^{j+1}$, then

$$\left([d]_1, \left[(d - D[L] \cdot \alpha^{L+1})\alpha^{N-L}\right]_2 \right) \approx_s \left(\left[\widetilde{r}\right]_1, \left[(\widetilde{r} - D[L] \cdot \alpha^{L+1})\alpha^{N-L}\right]_2 \right).$$

Therefore, $(\mathsf{crs}, \mathsf{com}, \pi_L)$ is statistically indistinguishable from $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{com}}, \widetilde{\pi}_{L,D[L]})$.

Location Binding and Malicious Security. We present the following theorems to claim the location binding and the malicious OW security.

Theorem 3. *Under* CDHE_N *assumption over* $\mathbb{G}_2$*, our construction satisfies the location binding property as defined in Section 4.1.*

Proof. Assume the adversary $\mathcal{A}$ can break the location binding property, then we can construct an algorithm $\mathcal{B}$ against the CDHE_N assumption over $\mathbb{G}_2$. Initially, $\mathcal{B}$ invokes an CDHE_N instance $(\mathbb{G}, \{[\alpha^j]_1\}_{j\in[N]}, \{[\alpha^j]_2\}_{j\in[2N]\setminus\{N+1\}})$, where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Then $\mathcal{B}$ proceeds as follows:

Setup. It directly returns $\mathsf{crs} := (\mathbb{G}, \{[\alpha^j]_1\}_{j\in[N]}, \{[\alpha^j]_2\}_{j\in[2N]\setminus\{N+1\}})$ to $\mathcal{A}$.

Challenge. Afterwards, it receives $\left(\mathsf{com}, L, \pi'_{L,0}, \pi'_{L,1}\right)$ from $\mathcal{A}$. If $(\pi'_{L,0}, \pi'_{L,1})$ are valid opening proofs at the same location L, then we have

$$\begin{aligned} & e([d]_1, [\alpha^{N-L}]_2) = e([1]_1, \pi'_{L,0}),\ e([d - \alpha^{L+1}]_1, [\alpha^{N-L}]_2) = e([1]_1, \pi'_{L,1}) \\ \Longrightarrow\ & e([\alpha^{L+1}]_1, [\alpha^{N-L}]_2) = e([1]_1, \pi'_{L,0}/\pi'_{L,1}) \end{aligned}$$

Equivalently, it holds that $\pi'_{L,0}/\pi'_{L,1} = [\alpha^{N+1}]_2$, thus breaking the CDHE_N assumption. Therefore, we complete the proof of this theorem.

Theorem 4. *Under* CBDHE_N *assumption, our construction satisfies the malicious OW security as defined in Section 4.2.*

Proof. We show that the adversary $\mathcal{A}$ cannot recover both m_0 and m_1, as otherwise we can build an efficient algorithm $\mathcal{B}$ against the CBDHE_N assumption.

Here, $\mathcal{B}$ needs to simulate the malicious WE scheme for an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. $\mathcal{B}$ invokes an CBDHE_N instance $(\mathbb{G}, \{[\alpha^j]_1\}_{j\in[N]}, \{[\alpha^j]_2\}_{j\in[2N]\setminus\{N+1\}}, [s]_1)$ where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Then $\mathcal{B}$ proceeds as follows:

Setup. It directly returns $\mathsf{crs} = (\mathbb{G}, \{[\alpha^j]_1\}_{j\in[N]}, \{[\alpha^j]_2\}_{j\in[2N]\setminus\{N+1\}})$.

Challenge. Upon receiving $(\mathsf{com}, L, \mathsf{st})$ from $\mathcal{A}_1$, it samples the shared randomness $s \leftarrow \mathbb{Z}_q$ and two random elements $C_0 \leftarrow \mathbb{G}_T$ and $C_1 \leftarrow \mathbb{G}_T$. Return $\mathsf{ct}_0 = ([s]_1, C_0)$ and $\mathsf{ct}_1 = ([s]_1, C_1)$. Note that these two ciphertexts are distributed identically to the real game defined in Section 4.2.

Then $\mathcal{A}_2$ outputs $(\mathsf{m}_0, \mathsf{m}_1)$. $\mathcal{B}$ returns $\frac{C_0\cdot\mathsf{m}_1}{C_1\cdot\mathsf{m}_0}$.

If $(\mathsf{m}_0, \mathsf{m}_1)$ are correct, then it would have

$$\frac{C_0}{\mathsf{m}_0} = [sd\alpha^{N-L}]_T, \quad \frac{C_1}{\mathsf{m}_1} = [sd\alpha^{N-L} - s\alpha^{N+1}]_T$$

Thus, $\frac{C_0\cdot\mathsf{m}_1}{C_1\cdot\mathsf{m}_0} = [s\alpha^{N+1}]$ is exactly a solution to CBDHE problem. If $\mathcal{A}_2$ recovers both m_0 and m_1 with non-negligible probability, $\mathcal{B}$ will break the CBDHE_N assumption with the same probability. Thus, we complete the proof. □

4.5 The Construction from CBDH

First, we show the equivocal vector commitment adapted from [CF13] as follows:

- $\mathsf{Setup}(1^\lambda, N)$: Generate $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample $v_1, \ldots, v_N \leftarrow \mathbb{Z}_q$ and output

$$\mathsf{crs} = (\mathbb{G}, \{[v_i]_1\}_{i\in[N]}, \{[v_i]_2, [v_i v_j]_2\}_{i\in[N], j\neq i}).$$

- $\mathsf{Com}(\mathsf{crs}, D)$: Parse $D = D[0]\|\cdots\|D[N-1] \in \{0,1\}^N$. Sample $r \leftarrow \mathbb{Z}_q$ and output $\mathsf{aux} = (D, r)$ and

$$\mathsf{com} = \left[\overbrace{r + \sum_{j=0}^{N-1} D[j]\cdot v_{j+1}}^{d}\right]_1.$$

- $\mathsf{Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L)$: Output $\pi_L = \left[r v_{L+1} + v_{L+1}\sum_{j=0, j\neq L}^{N-1} D[j]\cdot v_{j+1}\right]_2$.
- $\mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, b, \pi_L)$: Parse $\mathsf{com} = [d]_1$ and check whether

$$e([d - b\cdot v_{L+1}]_1, [v_{L+1}]_2) \stackrel{?}{=} e([1]_1, \pi_L).$$

  If pass, output 1; otherwise, output 0.
- $\mathsf{TSetup}(1^\lambda, N)$: It generates $\widetilde{\mathsf{crs}}$ as the crs in Setup and output $\mathsf{td} = \{v_i\}_{i\in[N]}$.
- $\mathsf{TCom}(\widetilde{\mathsf{crs}}, \mathsf{td})$: Sample $\widetilde{r} \leftarrow \mathbb{Z}_q$ and output $\widetilde{\mathsf{com}} = [\widetilde{r}]_1$ and $\mathsf{st} = \widetilde{r}$.
- $\mathsf{Equiv}(\widetilde{\mathsf{crs}}, \mathsf{td}, \mathsf{st}, L, b)$: Output $\widetilde{\pi}_{L,b} = \left[(\widetilde{r} - b\cdot v_{L+1})v_{L+1}\right]_2$.

Next, we present a concrete malicious WE for above commitment scheme as follows:

– $\mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, b))$: Let $\mathsf{com} = [d]_1$ and $L \in [0, N-1]$. Sample $s \leftarrow \mathbb{Z}_q$ and output

$$\mathsf{ct} = \left([\underbrace{s}_{c}]_1, \underbrace{[sdv_{L+1} - b \cdot sv_{L+1}^2]_T \cdot \mathsf{m}}_{C} \right)$$

– $\mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L)$: Parse $\mathsf{ct} = ([c]_1, C)$. Output $\mathsf{m} = C/e([c]_1, \pi_L)$.

Correctness and Efficiency. For the correctness of commitment, if $b = D[L]$, then we have

$$\begin{aligned} e([1]_1, \pi_L) \cdot e([b \cdot v_{L+1}]_1, [v_{L+1}]_2) &= [rv_{L+1} + \sum_{j=0, j\neq L}^{N-1} D[j] \cdot v_{j+1} + D[L] \cdot v_{L+1}]_T \\ &= \left[rv_{L+1} + \sum_{j=0}^{N-1} D[j] \cdot v_{j+1} \right]_T = [dv_{L+1}]_T , \end{aligned}$$

so the verification will pass and the commitment scheme is perfectly correct.

For the correctness of encryption, if $b = D[L]$, then we have $e([c_0]_1, \pi_L) = [srv_{L+1} + sv_{L+1} \sum_{j=0}^{N-1} D[j] \cdot v_{j+1} - D[L] \cdot sv_{L+1}^2]_T = [sdv_{L+1} - D[L] \cdot sv_{L+1}^2]_T = C/\mathsf{m}$. Therefore, $C/e([c_0]_1, \pi_L)$ exactly equals m, so the encryption scheme is perfectly correct.

For efficiency, assuming each group element of size $O(\lambda)$, we can see that $|\mathsf{com}| = O(\lambda), |\mathsf{ct}| = O(\lambda)$ and $(\mathsf{Ver}, \mathsf{Enc})$ only involves a constant number of exponent and pairing operations, which is independent of N.

Equivocality. Apparently, if we assume $r = \widetilde{r} - \sum_{j=0}^{N-1} D[j] \cdot v_{j+1}$, then

$$([d]_1, [(d - D[L] \cdot v_{L+1})v_{L+1}]_2) \approx_s \left([\widetilde{r}]_1, [(\widetilde{r} - D[L] \cdot v_{L+1})v_{L+1}]_2 \right).$$

Therefore, $(\mathsf{crs}, \mathsf{com}, \pi_L)$ is statistically indistinguishable from $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{com}}, \widetilde{\pi}_{L,D[L]})$.

Location Binding and Malicious Security. We present the following theorems to claim the location binding and the malicious OW security.

Theorem 5. *Under* CDH *assumption over* $\mathbb{G}_2$*, our construction satisfies the location binding property as defined in Section 4.1.*

Proof. Assume the adversary $\mathcal{A}$ can break the location binding property, then we can construct an algorithm $\mathcal{B}$ against the CDH assumption over $\mathbb{G}_2$. Initially, $\mathcal{B}$ invokes an CDH instance $(\mathbb{G}, [v]_1, [v]_2)$, where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Then $\mathcal{B}$ proceeds as follows:

Setup. It samples a random location $L' \leftarrow [0, N-1]$ that $\mathcal{A}$ will choose, and implicitly sets $v_{L'+1} = v$. Then sample other $v_i \leftarrow \mathbb{Z}_q$ for all $i \in [N] \setminus \{L'+1\}$ and return $\mathsf{crs} = (\mathbb{G}, \{[v_i]_1\}_{i\in[N]}, \{[v_i]_2, [v_i v_j]_2\}_{i\in[N], j\neq i})$ to $\mathcal{A}$.

Challenge. $\mathcal{B}$ receives $\left(\mathsf{com}, L, \pi'_{L,0}, \pi'_{L,1}\right)$ from $\mathcal{A}$. Here, if $L \neq L'$, $\mathcal{B}$ aborts; otherwise, as $(\pi'_{L,0}, \pi'_{L,1})$ are valid opening proofs at the same location L, then we have

$$\begin{aligned} & e([d]_1, [v_{L+1}]_2) = e([1]_1, \pi'_{L,0}),\ e([d - v_{L+1}]_1, [v_{L+1}]_2) = e([1]_1, \pi'_{L,1}) \\ \Longrightarrow\ & e([v_{L+1}]_1, [v_{L+1}]_2) = e([1]_1, \pi'_{L,0}/\pi'_{L,1}) \end{aligned}$$

Equivalently, it holds that $\pi'_{L,0}/\pi'_{L,1} = [v_{L+1}^2]_2$, thus breaking the CDH assumption. Therefore, we complete the proof of this theorem.

Theorem 6. *Under* CBDH *assumption, our construction satisfies the malicious OW security as defined in Section 4.2.*

Proof. We show that the adversary $\mathcal{A}$ cannot recover both m_0 and m_1, as otherwise we can build an efficient algorithm $\mathcal{B}$ against the CBDH assumption.

Here, $\mathcal{B}$ needs to simulate the malicious WE scheme for $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. Initially, $\mathcal{B}$ invokes an CBDH instance $(\mathbb{G}, [s]_1, [v]_1, [v]_2)$ where $\mathbb{G} := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Then $\mathcal{B}$ proceeds as follows:

Setup. It samples a random location $L' \in [0, N-1]$ and implicitly sets $v_{L'+1} = v$. Then sample other $v_i \leftarrow \mathbb{Z}_p$ for all $i \in [N] \setminus \{L'+1\}$ and return $\mathsf{crs} = (\mathbb{G}, \{[v_i]_1\}_{i\in[N]}, \{[v_i]_2, [v_i v_j]_2\}_{i\in[N], j\neq i})$.

Challenge. Upon receiving $(\mathsf{com}, L, \mathsf{st})$ from $\mathcal{A}_1$, it aborts if $L \neq L'$; otherwise, it samples the shared randomness $s \leftarrow \mathbb{Z}_q$ and two random elements $C_0 \leftarrow \mathbb{G}_T$ and $C_1 \leftarrow \mathbb{G}_T$. Return $\mathsf{ct}_0 = ([s]_1, C_0)$ and $\mathsf{ct}_1 = ([s]_1, C_1)$. These ciphertexts are distributed identically to the real game defined in Section 4.2.

Then $\mathcal{A}_2$ outputs $(\mathsf{m}_0, \mathsf{m}_1)$. $\mathcal{B}$ returns $\frac{C_0 \cdot \mathsf{m}_1}{C_1 \cdot \mathsf{m}_0}$.

If $(\mathsf{m}_0, \mathsf{m}_1)$ are correct, then it would have

$$\frac{C_0}{\mathsf{m}_0} = [sdv_{L+1}]_T, \quad \frac{C_1}{\mathsf{m}_1} = [sdv_{L+1} - sv_{L'+1}^2]_T$$

Thus, $\frac{C_0 \cdot \mathsf{m}_1}{C_1 \cdot \mathsf{m}_0} = [sv_{L'+1}^2]$ is exactly a solution to CBDH problem. If $\mathcal{A}_2$ recovers both m_0 and m_1 with non-negligible probability, $\mathcal{B}$ will break the CBDH assumption with the same probability. Thus, we complete the proof. □

4.6 The Construction from DLWE

First, we show the equivocal vector commitment adapted from the SIS-based Merkle hash as follows:

- $\mathsf{Setup}(1^\lambda, N)$: Sample $\mathbf{A}_0, \mathbf{A}_1, \mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{v}_0, \mathbf{v}_1 \leftarrow \mathbb{Z}_q^n$, and output $\mathsf{crs} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c)$, where we suppose the vector length $N = 2^\ell$ and c is a constant.
- $\mathsf{Com}(\mathsf{crs}, D)$: Parse $D = D[0] \| \cdots \| D[N-1] \in \{0,1\}^N$. Then it proceeds as follows:
  - For all $\mathsf{ind} \in \{0,1\}^\ell$, sample $\mathbf{x}_{\mathsf{ind}} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_0}^m$ and set
  $$\mathbf{y}_{\mathsf{ind}||D[\mathsf{ind}]} = \mathbf{B}\mathbf{x}_{\mathsf{ind}} \in \mathbb{Z}_q^n, \ \mathbf{y}_{\mathsf{ind}||\overline{D}[\mathsf{ind}]} = \mathbf{v}_{\overline{D}[\mathsf{ind}]} \in \mathbb{Z}_q^n.$$
  - For all $j = \ell, \ldots, 0$, sample $\mathbf{r}_{\mathsf{ind}\|0}, \mathbf{r}_{\mathsf{ind}\|1} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_0}^{m'}$ for all all $\mathsf{ind} \in \{0,1\}^j$. Then use these randomness to recursively compute
  $$\mathbf{y}_{\mathsf{ind}} = \mathbf{A}_0\overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}||0}) + \mathbf{A}_1\overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}||1}) \in \mathbb{Z}_q^n, \ \forall \mathsf{ind} \in \{0,1\}^j$$

  Output $\mathsf{com} = \mathbf{y}_\epsilon \in \mathbb{Z}_q^n$ and $\mathsf{aux} = (\{\mathbf{x}_{\mathsf{ind}}\}_{\mathsf{ind}\in\{0,1\}^\ell}, \{\mathbf{r}_{\mathsf{ind}}, \mathbf{y}_{\mathsf{ind}}\}_{j\in[0,\ell],\mathsf{ind}\in\{0,1\}^j})$.
- $\mathsf{Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L)$: Output $\pi_L = (\mathbf{x}_L, \{\mathbf{r}_{L[:j]\|\beta}, \mathbf{y}_{L[:j]\|\beta}\}_{j\in[0,\ell],\beta\in\{0,1\}})$.
- $\mathsf{Ver}(\mathsf{crs}, \mathsf{com}, L, b, \pi_L)$: Parse $\mathsf{com} = \mathbf{y}_\epsilon$. Check whether
$$\mathbf{y}_{L||b} \stackrel{?}{=} \mathbf{B}\mathbf{x}_{\mathsf{ind}}, \quad \mathbf{y}_{L||\overline{b}} \stackrel{?}{=} \mathbf{v}_{\overline{b}},$$
$$\mathbf{y}_{L[:j-1]} \stackrel{?}{=} (\mathbf{A}_0 \mid \mathbf{A}_1)\begin{bmatrix}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{L[:j-1]||0}) \\ \overline{\mathbf{G}}^{-1}(\mathbf{y}_{L[:j-1]||1})\end{bmatrix}, \ \forall j \in [\ell+1].$$

  If pass, output 1; otherwise, output 0.
- $\mathsf{TSetup}(1^\lambda, N)$: Sample
$$\widehat{\mathbf{A}}_0 \leftarrow \mathbb{Z}_q^{n\times m}, \ \mathbf{v}_0, \mathbf{v}_1 \leftarrow \mathbb{Z}_q^n,$$
$$(\widehat{\mathbf{A}}_1, \mathbf{T}_{\widehat{\mathbf{A}}_1}) \leftarrow \mathsf{TGen}(1^n, 1^m, q), \ (\mathbf{B}, \mathbf{T}_{\mathbf{B}}) \leftarrow \mathsf{TGen}(1^n, 1^m, q).$$

  Then for $b \in \{0,1\}$, we let $\widehat{\mathbf{A}}_b = (\widehat{\mathbf{A}}_{b,0} \mid \widehat{\mathbf{A}}_{b,1})$ and set $\mathbf{A}_b = (\widehat{\mathbf{A}}_{0,b} \mid \widehat{\mathbf{A}}_{1,b})$. Output $\widetilde{\mathsf{crs}} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c)$ and $\mathsf{td} = (\mathbf{T}_{\widehat{\mathbf{A}}_1}, \mathbf{T}_{\mathbf{B}})$, where we suppose $N = 2^\ell$ and c is a constant.

– $\mathsf{TCom}(\widetilde{\mathsf{crs}}, \mathsf{td})$: It proceeds as follows:
  - Sample $\mathbf{y}_{\mathsf{ind}} \leftarrow \mathbb{Z}_q^n$ for all $\mathsf{ind} \in \{0,1\}^\ell$.
  - For all $j = \ell - 1, ..., 0$, sample $\mathbf{r}_{\mathsf{ind}\|0}, \mathbf{r}_{\mathsf{ind}\|1} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_0}^{m'}$ for all $\mathsf{ind} \in \{0,1\}^j$. Then it uses the randomness to recursively compute

$$\mathbf{y}_{\mathsf{ind}} = \mathbf{A}_0\overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}||0}) + \mathbf{A}_1\overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}||1}) \in \mathbb{Z}_q^n, \ \forall \mathsf{ind} \in \{0,1\}^j$$

  Output $\widetilde{\mathsf{com}} = \mathbf{y}_\epsilon \in \mathbb{Z}_q^n$ and $\mathsf{st} = (\{\mathbf{r}_{\mathsf{ind}}, \mathbf{y}_{\mathsf{ind}}\}_{j\in[\ell],\mathsf{ind}\in\{0,1\}^j})$.
– $\mathsf{Equiv}(\widetilde{\mathsf{crs}}, \mathsf{td}, \mathsf{st}, L, b)$: Parse $\mathsf{st} = (\{\mathbf{r}_{\mathsf{ind}}, \mathbf{y}_{\mathsf{ind}}\}_{j\in[\ell],\mathsf{ind}\in\{0,1\}^j})$. We rewrite $\mathbf{y}_L$ into the following form:

$$\begin{aligned}\mathbf{y}_L &= \mathbf{A}_b\overline{\mathbf{G}}^{-1}(\mathbf{y}_{L\|b}) + \mathbf{A}_{\bar{b}}\overline{\mathbf{G}}^{-1}(\mathbf{v}_{\bar{b}}) \\ &= (\widehat{\mathbf{A}}_{0,b} \mid \widehat{\mathbf{A}}_{1,b})\overline{\mathbf{G}}^{-1}(\mathbf{y}_{L\|b}) + (\widehat{\mathbf{A}}_{0,\bar{b}} \mid \widehat{\mathbf{A}}_{1,\bar{b}})\overline{\mathbf{G}}^{-1}(\mathbf{v}_{\bar{b}}) \\ &= (\widehat{\mathbf{A}}_{0,b} \mid \widehat{\mathbf{A}}_{1,b})\begin{pmatrix}\mathbf{G}^{-1}(\mathbf{y}_{L\|b}) \\ \mathbf{r}_{L\|b}\end{pmatrix} + (\widehat{\mathbf{A}}_{0,\bar{b}} \mid \widehat{\mathbf{A}}_{1,\bar{b}})\begin{pmatrix}\mathbf{G}^{-1}(\mathbf{v}_{\bar{b}}) \\ \mathbf{r}_{L\|\bar{b}}\end{pmatrix}.\end{aligned}$$

  Then it is equivalent to

$$\underbrace{(\widehat{\mathbf{A}}_{1,b} \mid \widehat{\mathbf{A}}_{1,\bar{b}})}_{\widehat{\mathbf{A}}_1}\begin{pmatrix}\mathbf{r}_{L\|b} \\ \mathbf{r}_{L\|\bar{b}}\end{pmatrix} = \underbrace{\mathbf{y}_L - \widehat{\mathbf{A}}_{0,\bar{b}}\mathbf{G}^{-1}(\mathbf{v}_{\bar{b}}) - \widehat{\mathbf{A}}_{0,b}\mathbf{G}^{-1}(\mathbf{y}_{L\|b})}_{\mathbf{z}_L}$$

  With the trapdoor $\mathsf{td} = (\mathbf{T}_{\widehat{\mathbf{A}}_1}, \mathbf{T}_{\mathbf{B}})$, it randomly samples $\mathbf{y}_{L||b} \leftarrow \mathbb{Z}_q^n$ and then compute

$$\begin{aligned}(\mathbf{r}_{L\|b}, \mathbf{r}_{L\|\bar{b}}) &\leftarrow \mathsf{SamplePre}(\widehat{\mathbf{A}}_1, \mathbf{T}_{\widehat{\mathbf{A}}_1}, \mathbf{z}_L, \sigma_0), \\ \mathbf{x}_{L,b} &\leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{T}_{\mathbf{B}}, \mathbf{y}_{L||b}, \sigma_0).\end{aligned}$$

  Therefore, it outputs $\widetilde{\pi}_{L,b} = (\mathbf{x}_{L,b}, \{\mathbf{r}_{L[:j]\|\beta}, \mathbf{y}_{L[:j]\|\beta}\}_{j\in[0,\ell],\beta\in\{0,1\}})$.

Next, we present a concrete malicious WE for above commitment scheme as follows:

– $\mathsf{Enc}(\mathsf{crs}, (\mathsf{com}, L, b), \mathsf{m})$: Parse $\mathsf{crs} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c)$, $\mathsf{com} = \mathbf{y}_\epsilon$ and $L = L[0]\| \ldots \|L[\ell-1] \in \{0,1\}^\ell$. Then it proceeds as follows:
  - For all $i \in [\lambda]$, sample randomness $\mathbf{t}_i = (t_{i,0}, t_{i,1}) \leftarrow \mathbb{Z}_q^2, \mathbf{s}_i = (\mathbf{s}_{i,0}, \ldots, \mathbf{s}_{i,\ell}, \mathbf{s}_{i,\ell+1}) \leftarrow \mathbb{Z}_q^{(\ell+2)n}$ and $\mathbf{e}_i = (e_{i,0}, e_{i,1}, \mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,\ell+2}) \leftarrow \mathcal{D}_{\mathbb{Z},\sigma_1}^{2+(2\ell+3)m}$.
  - Compute

$$\begin{aligned}\mathbf{c}_{i,j}^\top &= -\mathbf{s}_{i,j-1}^\top(\mathbf{A}_0 \mid \mathbf{A}_1) + \mathbf{s}_{i,j}^\top(\overline{L}[j-1]\cdot\overline{\mathbf{G}} | L[j-1]\cdot\overline{\mathbf{G}}) + \mathbf{e}_{i,j}^\top, \forall j \in [\ell] \\ \mathbf{c}_{i,\ell+1}^\top &= -\mathbf{s}_{i,\ell}^\top(\mathbf{A}_0 \mid \mathbf{A}_1) + \mathbf{s}_{i,\ell+1}^\top(\overline{\mathbf{G}} \mid \overline{\mathbf{G}}) + \mathbf{e}_{i,\ell+1}^\top, \\ \mathbf{c}_{i,\ell+2}^\top &= -\mathbf{s}_{i,\ell+1}^\top\mathbf{B} + \mathbf{e}_{i,\ell+2}^\top, \\ k_i &= \lfloor -\mathbf{s}_{i,0}^\top\mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top\mathbf{v}_{\bar{b}} + e_{i,b} + t_{i,b}\rfloor_c.\end{aligned}$$

  - Let $\mathsf{k} = k_1||\cdots||k_\lambda$ and compute $C = \mathsf{m} \oplus \mathsf{k}$.

  Output $\mathsf{ct} = (\{t_{i,b}, \mathbf{c}_{i,j}\}_{i\in[\lambda],j\in[\ell+2]}, C)$.
– $\mathsf{Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, b), \pi_L)$: Parse $L = L[0]\| \ldots \|L[\ell-1] \in \{0,1\}^\ell$ and $\pi_L = (\mathbf{x}_L, \{\mathbf{r}_{L[:j]\|\beta}, \mathbf{y}_{L[:j]\|\beta}\}_{j\in[0,\ell],\beta\in\{0,1\}})$. For all $i \in [\lambda]$, compute

$$k_i = \left\lfloor \sum_{j=1}^{\ell+1} \mathbf{c}_{i,j}^\top \overbrace{\begin{bmatrix}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,L[:j-1]||0}) \\ \overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,L[:j-1]||1})\end{bmatrix}}^{\mathbf{w}_{i,j-1}} + \mathbf{c}_{i,\ell+2}^\top\mathbf{x}_{i,L} + t_{i,D[L]} \right\rfloor_c$$

  Let $\mathsf{k} = k_1||\cdots||k_\lambda$ and output $\mathsf{m} = C \oplus \mathsf{k}$.

Parameters and Efficiency. For parameter choices, we set $n, m, m', c, q, \sigma_0, \sigma_1$ as follows:

$$m = 2n\log q, \quad \sigma_0 = m\log n, \quad m' = n\log q, \quad c = 4,$$
$$q \geq c\lambda^3\left(2(m\sqrt{\lambda}\sigma_0 + (\ell+1)(m + m\sqrt{\lambda}\sigma_0) + 1)\sqrt{\lambda}\sigma_1 + 1\right)$$

where (n, σ_1) is implied by $O(\lambda)$. Therefore, we have $|\mathsf{com}| = O(\lambda)$, $|\mathsf{ct}| = O(\lambda^2 \log N)$ and the runtime of Enc is at most linear in $\log N$.

Correctness. For the correctness of commitment, we can see that π_L is indeed a path of the Merkle hash tree from the top node to the leaf node corresponding to database bit $D[L]$, so the verification for π_L will naturally pass and the commitment scheme is perfectly correct. Next, we show that the above WE scheme is $1/\lambda^2$-correct. Recall that $\mathbf{y}_{i,L||D[L]} = \mathbf{B}\mathbf{x}_L$ and $\mathbf{y}_{i,L||\overline{D}[L]} = \mathbf{v}_{i,\overline{D}[L]}$. For all $j = \ell, ..., 0$, we have

$$\mathbf{y}_{i,\mathsf{ind}} = \mathbf{A}_{i,0}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,\mathsf{ind}||0}) + \mathbf{A}_{i,1}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,\mathsf{ind}||1}),\ \forall \mathsf{ind} \in \{0,1\}^j$$

With above equality, for all $i \in [\lambda]$ we have

$$\begin{aligned}
k_i &= \left\lfloor \sum_{j=1}^{\ell} \mathbf{c}_{i,j}^\top \mathbf{w}_{i,j-1} + \mathbf{c}_{i,\ell+1}^\top \mathbf{w}_{i,\ell} + \mathbf{c}_{i,\ell+2}^\top \mathbf{x}_{i,L} + t_{i,D[L]} \right\rfloor_c \\
&= \left\lfloor \sum_{j=1}^{\ell} \left(-\mathbf{s}_{i,j-1}^\top \left(\mathbf{A}_{i,0}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,L[:j-1]||0}) + \mathbf{A}_{i,1}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,L[:j-1]||1})\right) + \mathbf{s}_{i,j}^\top \mathbf{y}_{i,L[:j]}\right)\right. \\
&\quad + \left(-\mathbf{s}_{i,\ell}^\top \left(\mathbf{A}_{i,0}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,L||0}) + \mathbf{A}_{i,1}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{i,L||1})\right) + \mathbf{s}_{i,\ell+1}^\top(\mathbf{y}_{i,L||0} + \mathbf{y}_{i,L||1})\right) \\
&\quad \left. - \mathbf{s}_{i,\ell+1}^\top \mathbf{B}\mathbf{x}_{i,L} + \mathbf{e}_{i,\ell+2}^\top \mathbf{x}_{i,L} + \sum_{j=1}^{\ell+1} \mathbf{e}_{i,j}^\top \mathbf{w}_{i,j-1} + t_{i,D[L]} \right\rfloor_c \\
&= \left\lfloor \sum_{j=1}^{\ell} \left(-\mathbf{s}_{i,j-1}^\top \mathbf{y}_{i,L[:j-1]} + \mathbf{s}_{i,j}^\top \mathbf{y}_{i,L[:j]}\right) - \mathbf{s}_{i,\ell}^\top \mathbf{y}_{i,L} + \mathbf{s}_{i,\ell+1}^\top(\mathbf{y}_{i,L||D[L]} + \mathbf{y}_{i,L||\overline{D}[L]})\right. \\
&\quad \left. - \mathbf{s}_{i,\ell+1}^\top \mathbf{y}_{i,L||D[L]} + \mathbf{e}_{i,\ell+2}^\top \mathbf{x}_{i,L} + \sum_{j=1}^{\ell+1} \mathbf{e}_{i,j}^\top \mathbf{w}_{i,j-1} + t_{i,D[L]} \right\rfloor_c \\
&= \left\lfloor -\mathbf{s}_{i,0}^\top \mathbf{y}_{i,\epsilon} + \mathbf{s}_{i,\ell+1}^\top \mathbf{y}_{i,L||\overline{D}[L]} + \mathbf{e}_{i,\ell+2}^\top \mathbf{x}_{i,L} + \sum_{j=1}^{\ell+1} \mathbf{e}_{i,j}^\top \mathbf{w}_{i,j-1} + t_{i,D[L]} \right\rfloor_c \\
&= \left\lfloor -\mathbf{s}_{i,0}^\top \mathbf{y}_{i,\epsilon} + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_{i,\overline{D}[L]} + e_{i,D[L]} + \underbrace{\mathbf{e}_{i,\ell+2}^\top \mathbf{x}_{i,L} + \sum_{j=1}^{\ell+1} \mathbf{e}_{i,j}^\top \mathbf{w}_{i,j-1} - e_{i,D[L]}}_{e_i} + t_{i,D[L]} \right\rfloor_c \\
&= \left\lfloor -\mathbf{s}_{i,0}^\top \mathbf{y}_{i,\epsilon} + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_{i,\overline{D}[L]} + e_{i,D[L]} + t_{i,D[L]} \right\rfloor_c \\
&= k_{i,D[L]}
\end{aligned}$$

Note that for the last second equality, as $t_{i,D[L]}$ is randomly sampled, we can remove the error term e_i by lemma 3. Concretely, observe that $m = 2n\log q$ and by lemma 1, e_i is bounded by

$$B = (m\sqrt{\lambda}\sigma_0 + (\ell+1)(m + m\sqrt{\lambda}\sigma_0) + 1)\sqrt{\lambda}\sigma_1 \approx O(\lambda \ell m^2 \sigma_1),$$

so the last-second equality holds with probability $1 - (2B+1)\cdot c/q$. By setting $q \geq c\lambda^3(2B+1)$, the equality holds with at least probability $1 - 1/\lambda^3$. Thus, it can properly recover all keys $k_{1,D[L]}, \ldots, k_{\lambda,D[L]}$ except with probability

$1/\lambda^2$. Since the ciphertext includes $C = \mathsf{m} \oplus (k_{1,D[L]} \| \cdots \| k_{\lambda,D[L]})$, then it can recover m with the same probability. Therefore, the correctness follows readily.

Equivocality. Observe that for all $\mathsf{ind} \in \{0,1\}^\ell$,

$$\begin{aligned}
\mathbf{y}_{\mathsf{ind}} &= \mathbf{A}_{D[\mathsf{ind}]}\overline{\mathbf{G}}^{-1}(\mathbf{y}_{\mathsf{ind}\|D[L]}) + \mathbf{A}_{\overline{D}[\mathsf{ind}]}\overline{\mathbf{G}}^{-1}(\mathbf{v}_{\overline{D}[\mathsf{ind}]}) \\
&= (\widehat{\mathbf{A}}_{0,D[\mathsf{ind}]} \mid \widehat{\mathbf{A}}_{1,D[\mathsf{ind}]}) \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{y}_{\mathsf{ind}\|D[L]}) \\ \mathbf{r}_{\mathsf{ind}\|D[L]} \end{pmatrix} + (\widehat{\mathbf{A}}_{0,\overline{D}[\mathsf{ind}]} \mid \widehat{\mathbf{A}}_{1,\overline{D}[\mathsf{ind}]}) \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{v}_{\overline{D}[\mathsf{ind}]}) \\ \mathbf{r}_{\mathsf{ind}\|\overline{D}[L]} \end{pmatrix} \\
&= (\widehat{\mathbf{A}}_{0,D[\mathsf{ind}]} \mid \widehat{\mathbf{A}}_{0,\overline{D}[\mathsf{ind}]}) \begin{pmatrix} \mathbf{G}^{-1}(\mathbf{y}_{\mathsf{ind}\|D[L]}) \\ \mathbf{G}^{-1}(\mathbf{v}_{\overline{D}[\mathsf{ind}]}) \end{pmatrix} + \underbrace{(\widehat{\mathbf{A}}_{1,D[\mathsf{ind}]} \mid \widehat{\mathbf{A}}_{1,\overline{D}[\mathsf{ind}]}) \begin{pmatrix} \mathbf{r}_{\mathsf{ind}\|D[L]} \\ \mathbf{r}_{\mathsf{ind}\|\overline{D}[L]} \end{pmatrix}}_{\mathbf{u}_{\mathsf{ind}}}.
\end{aligned}$$

Since both $\mathbf{r}_{\mathsf{ind}\|D[L]}$ and $\mathbf{r}_{\mathsf{ind}\|\overline{D}[L]}$ are randomly sampled from $\mathcal{D}^{m'}_{\mathbb{Z},\sigma_0}$ (where $m' = n\log q$), by lemma 2, we can see that $\mathbf{u}_{\mathsf{ind}}$ is statistically close to the uniform distribution. In addition, we can see that $(\mathbf{r}_{L\|b}, \mathbf{r}_{L\|\bar{b}})$ and $\mathbf{x}_{L,b}$ are statistically close to the distributions $\mathcal{D}^{m'}_{\mathbb{Z},\sigma_0}$ and $\mathcal{D}^{m}_{\mathbb{Z},\sigma_0}$, respectively. Therefore, $(\mathsf{crs}, \mathsf{com}, \pi_L)$ is statistically indistinguishable from $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{com}}, \widetilde{\pi}_{L,D[L]})$.

Location Binding and Malicious Security. We present the following theorems to claim the location binding and the malicious OW security.

Theorem 7. *Under* $\mathsf{SIS}_{n,2m,q,\sigma_0\sqrt{\lambda m}}$ *and inhomogeneous* $\mathsf{SIS}_{n,m,q,\sigma_0\sqrt{\lambda m}}$ *assumptions, our construction satisfies the location binding as defined in Section 4.1.*

Proof. Assume the adversary $\mathcal{A}$ can break the location binding property, then we can construct an algorithm $\mathcal{B}$ against either $\mathsf{SIS}_{n,2m,q,\delta}$ assumption or inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ assumption, where $\delta = \sigma_0\sqrt{m\lambda}$ and $m = 2n\log q$. Here, since our WE scheme sets $n = O(\lambda)$ and $q > \lambda^3\sigma_0 m\sqrt{\lambda} > \lambda^3\delta$, both $\mathsf{SIS}_{n,2m,q,\delta}$ and inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ are guaranteed to be hard, according to the analysis of [GPV08].

Initially, $\mathcal{B}$ invokes an $\mathsf{SIS}_{n,2m,q,\delta}$ instance $(\mathbf{A}_0\|\mathbf{A}_1)$, where $\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathbb{Z}_q^{n\times m}$. Then it invokes an inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ instance $(\mathbf{B}, \mathbf{v})$, where $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{v}_1 \leftarrow \mathbb{Z}_q^n$. Then $\mathcal{B}$ proceeds as follows:

Setup. It implicitly sets $\mathbf{v}_1 = \mathbf{v}$ and samples $\mathbf{v}_0 \leftarrow \mathbb{Z}_q^n$. Return $\mathsf{crs} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c)$ to $\mathcal{A}$.

Challenge. $\mathcal{B}$ receives $\left(\mathsf{com}, L, \pi'_{L,0}, \pi'_{L,1}\right)$ from $\mathcal{A}$. Let $\pi'_{L,0} = (\mathbf{x}'_{L,0}, \{\mathbf{r}'_{L[:j]\|\beta,0}, \mathbf{y}'_{L[:j]\|\beta,0}\}_{j\in[0,\ell],\beta\in\{0,1\}})$ and $\pi'_{L,1} = (\mathbf{x}'_{L,1}, \{\mathbf{r}'_{L[:j]\|\beta,1}, \mathbf{y}'_{L[:j]\|\beta,1}\}_{j\in[0,\ell],\beta\in\{0,1\}})$, then there must exist some $j \in [0, \ell-1]$ such that $\mathbf{y}'_{L[:j],0} = \mathbf{y}'_{L[:j],1}$, i.e.,

$$\begin{aligned}
&(\mathbf{A}_0 \mid \mathbf{A}_1) \begin{pmatrix} \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,0}) \\ \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,0}) \end{pmatrix} = (\mathbf{A}_0 \mid \mathbf{A}_1) \begin{pmatrix} \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,0}) \\ \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,1}) \end{pmatrix} \\
\Longrightarrow\ &(\mathbf{A}_0 \mid \mathbf{A}_1) \underbrace{\begin{pmatrix} \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,0}) - \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,1}) \\ \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,0}) - \overline{\mathbf{G}}^{-1}(\mathbf{y}'_{L[:j]\|1,1}) \end{pmatrix}}_{\mathbf{z}} = \mathbf{0}
\end{aligned}$$

Next, we consider two cases:

- If $\mathbf{z} \neq \mathbf{0}$, then $\mathbf{z}$ can be viewed as a solution to the $\mathsf{SIS}_{n,2m,q,\delta}$ assumption [GPV08]. Thus, in this case, if $\mathcal{A}$ breaks the location binding with non-negligible probability, $\mathcal{B}$ can break the $\mathsf{SIS}_{n,2m,q,\delta}$ assumption with the same probability.
- If $\mathbf{z} = \mathbf{0}$, i.e., $\mathbf{y}'_{L[:j]\|b,0} = \mathbf{y}'_{L[:j]\|b,1}$ for all $j \in [0, \ell-1]$ and $b \in \{0,1\}$, then it must have $\mathbf{y}'_{L\|1,0} = \mathbf{v}_1 = \mathbf{B}\mathbf{x}'_{L,1} = \mathbf{y}'_{L\|1,1}$, while $\mathbf{v}_1$ actually belongs to the inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ instance. Thus, $\mathbf{x}'_{L,1}$ can be viewed as a solution to the inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ assumption. Namely, in this case, if $\mathcal{A}$ breaks the location binding with non-negligible probability, $\mathcal{B}$ can break the inhomogeneous $\mathsf{SIS}_{n,m,q,\delta}$ assumption with the same probability.

Above all, we complete the proof of this theorem.

Theorem 8. *Under matrix* $\mathsf{DLWE}_{n,2m,\lambda,q,\sigma_1}$ *and matrix* $\mathsf{DLWE}_{n,(m+1),\lambda,q,\sigma_1}$ *assumptions (given above parameters), our construction satisfies the malicious OW security as defined in Section 4.2.*

Proof. We show that the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ cannot recover both m_0 and m_1, as otherwise we can build an efficient algorithm $\mathcal{B}$ against the matrix DLWE assumption.
Game Sequence. We argue the security via the following games.

- G_0 : This is the real game. The challenger initializes

$$\mathsf{crs} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c).$$

  Then it receives $(\mathsf{com}, L, \mathsf{st})$ from the adversary, where $\mathsf{com} = y_\epsilon$, and generates the challenge ciphertexts in the following form:

$$\mathsf{ct}_b = (\{t_{i,b}, \mathbf{c}_{i,j}\}_{i\in[\lambda], j\in[\ell+2]}, C_b),\ \forall b \in \{0,1\}$$

  where $t_{i,b}$ is randomly sampled over $\mathbb{Z}_q^2$ and

$$\begin{aligned}
\mathbf{c}_{i,j}^\top &= -\mathbf{s}_{i,j-1}^\top(\mathbf{A}_0 \mid \mathbf{A}_1) + \mathbf{s}_{i,j}^\top(\overline{L}[j-1]\cdot\overline{\mathbf{G}} \mid L[j-1]\cdot\overline{\mathbf{G}}) + \mathbf{e}_{i,j}^\top, \forall j \in [\ell]\\
\mathbf{c}_{i,\ell+1}^\top &= -\mathbf{s}_{i,\ell}^\top(\mathbf{A}_0 \mid \mathbf{A}_1) + \mathbf{s}_{i,\ell+1}^\top(\overline{\mathbf{G}} \mid \overline{\mathbf{G}}) + \mathbf{e}_{i,\ell+1}^\top\\
\mathbf{c}_{i,\ell+2}^\top &= -\mathbf{s}_{i,\ell+1}^\top\mathbf{B} + \mathbf{e}_{i,\ell+2}^\top\\
C_b &= \mathsf{m}_b \oplus \mathsf{k}_b
\end{aligned}$$

  Here, we consider the shared randomness $\mathbf{s}_{i,0}, \ldots, \mathbf{s}_{i,\ell}, \mathbf{s}_{i,\ell+1} \leftarrow \mathbb{Z}_q^n$ and $e_{i,0}, e_{i,1} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}, \mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,\ell}, \mathbf{e}_{i,\ell+1} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{2m}, \mathbf{e}_{i,\ell+2} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{m}$, and $(\mathsf{m}_0, \mathsf{m}_1)$ are random messages over $\{0,1\}^{\lambda\log c}$. Besides, for all $i \in [\lambda]$, it computes

$$k_{i,b} = \lfloor -\mathbf{s}_{i,0}^\top\mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top\mathbf{v}_{\overline{b}} + e_{i,b} + t_{i,b}\rfloor_c$$

  and implicitly sets $\mathsf{k}_0 = k_{1,0}||\cdots||k_{\lambda,0}$, $\mathsf{k}_1 = k_{1,1}||\cdots||k_{\lambda,1}$.
  Upon receiving $(\mathsf{ct}_0, \mathsf{ct}_1)$, the adversary will guess the messages m_0 and m_1. It will win if the challenger gets the correct m_0 and m_1.
- G_1 : This game is identical to G_0 except that in the challenge phase, instead of sampling $(\mathsf{m}_0, \mathsf{m}_1)$, the challenger will directly sample $C_0, C_1 \leftarrow \{0,1\}^{\lambda\log c}$.
- G_2 : We define a weaker game that is identical to G_1 except that after receiving $(\mathsf{m}_0, \mathsf{m}_1)$ from $\mathcal{A}_2$, the challenger will directly set $\mathsf{k}_0 = \mathsf{m}_0 \oplus C_0 = k_{1,0}||\cdots||k_{\lambda,0}$ and $\mathsf{k}_1 = \mathsf{m}_1 \oplus C_1 = k_{1,0}||\cdots||k_{\lambda,0}$, then define the following sets

$$\begin{gathered}
S_{i,0} = \{x \in \mathbb{Z}_q \mid \lfloor x\rfloor_c = k_{i,0}\}, \quad S_{i,1} = \{x \in \mathbb{Z}_q \mid \lfloor x\rfloor_c = k_{i,1}\}\\
T_i = \{(x_0 - x_1) \in \mathbb{Z}_q \mid x_0 \in S_{i,0}, x_1 \in S_{i,1}\}
\end{gathered} \tag{4}$$

  and check whether $\mathbf{s}_{i,\ell+1}^\top(\mathbf{v}_1 - \mathbf{v}_0) + (e_{i,0} - e_{i,1}) + (t_{i,0} - t_{i,1}) \in T_i$ for all $i \in [\lambda]$. The adversary wins if all checks pass.
- $\mathsf{G}_{3,\kappa}(\kappa \in [\ell+1])$: This is identical to $\mathsf{G}_{3,\kappa-1}$ except that we replace $\mathbf{c}_{i,\kappa}$ with $\widetilde{\mathbf{c}}_{i,\kappa} \leftarrow \mathbb{Z}_q^{2m}$ for all $i \in [\lambda]$. This follows from Lemma 9.
- G_4 : This is identical to $\mathsf{G}_{3,\ell+1}$ except that we replace $\mathbf{c}_{i,\ell+2}$ with $\widetilde{\mathbf{c}}_{i,\ell+2} \leftarrow \mathbb{Z}_q^m$ for all $i \in [\lambda]$. In this case, we claim that the matrix DLWE assumption can be broken if the adversary can efficiently recover both $(\mathsf{k}_0, \mathsf{k}_1)$.

Lemma 7. *For any efficient adversary* $\mathcal{A}$*, it has* $\mathsf{G}_0 \approx_s \mathsf{G}_1$.

Proof. Observe that the only difference between G_0 and G_1 is the generation of (C_0, C_1). Since in both games the distribution of (C_0, C_1) is indistinguishable from the uniform distribution, it naturally has $\mathsf{G}_0 \approx_s \mathsf{G}_1$.

Lemma 8. *For any efficient adversaries $\mathcal{A}$ and $\mathcal{A}'$, let $\mathsf{Adv}^1_{\mathcal{A}}(\lambda)$ denote the advantage function of $\mathcal{A}$ in G_1 and $\mathsf{Adv}^2_{\mathcal{A}'}(\lambda)$ denote the advantage function of $\mathcal{A}'$ in G_2, then it has*

$$\mathsf{Adv}^1_{\mathcal{A}}(\lambda) \le \mathsf{Adv}^2_{\mathcal{A}'}(\lambda).$$

Proof. It suffices to show that the probability that $\mathcal{A}'$ wins in G_2 at least equals the probability that $\mathcal{A}$ wins in G_1, i.e., $\Pr[\mathcal{A}' \text{ WIN}] \ge \Pr[\mathcal{A} \text{ WIN}]$. Because

$$\begin{aligned}\Pr[\mathcal{A}' \text{ WIN}] &= \Pr[\mathcal{A}' \text{ WIN} \mid \mathcal{A} \text{ WIN}] \cdot \Pr[\mathcal{A} \text{ WIN}] \\ &\quad + \Pr[\mathcal{A}' \text{ WIN} \mid \neg(\mathcal{A} \text{ WIN})] \cdot \Pr[\neg(\mathcal{A} \text{ WIN})] \\ &\ge \Pr[\mathcal{A}' \text{ WIN} \mid \mathcal{A} \text{ WIN}] \cdot \Pr[\mathcal{A} \text{ WIN}]\end{aligned}$$

Next, we show that $\Pr[\mathcal{A}' \text{ WIN} \mid \mathcal{A} \text{ WIN}] = 1$. If the adversary correctly recovers both m_0 and m_1 (corresponding to the event $\mathcal{A}$ WIN), then $\mathsf{k}_0 = C_0 \oplus \mathsf{m}_0 = k_{1,0} \| \cdots \| k_{\lambda,0}$ and $\mathsf{k}_1 = C_1 \oplus \mathsf{m}_1 = k_{1,1} \| \cdots \| k_{\lambda,1}$ can also be recovered, so it would have

$$\begin{aligned}k_{i,0} &= \lfloor -\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_1 + e_{i,0} + t_{i,0} \rfloor_c \\ k_{i,1} &= \lfloor -\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_0 + e_{i,1} + t_{i,1} \rfloor_c\end{aligned}$$

where $i \in [\lambda]$. Thus, we can define the sets $R_{i,0}, R_{i,1}$ and T_i as in equality (4), and it must have $-\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_1 + e_{i,0} + t_{i,0} \in S_{i,0}$ and $-\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_0 + e_{i,1} + t_{i,1} \in S_{i,1}$. Evidently, it must hold that $\mathbf{s}_{i,\ell+1}^\top (\mathbf{v}_1 - \mathbf{v}_0) + (e_{i,0} - e_{i,1}) + (t_{i,0} - t_{i,1}) \in T_i$ (corresponding to the event $\mathcal{A}'$ WIN). Therefore, we complete the proof of this lemma. □

Lemma 9. *Under matrix $\mathsf{DLWE}_{n,2m,\lambda,q,\sigma_1}$ assumption, it has $\mathsf{G}_{3,\kappa-1} \approx_c \mathsf{G}_{3,\kappa}$.*

Proof. Assume an algorithm $\mathcal{B}$ against matrix $\mathsf{DLWE}_{n,2m,\lambda,q,\sigma_1}$ assumption and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the malicious OW security of WE. Initially, $\mathcal{B}$ invokes a matrix DLWE instance $(\mathbf{A}, \mathbf{U})$ where $\mathbf{A} = (\mathbf{A}_0 \mid \mathbf{A}_1) \in \mathbb{Z}_q^{n \times 2m}$ and $\mathbf{U} = ((\mathbf{u}_{1,0}, \mathbf{u}_{1,1}) \mid \cdots \mid (\mathbf{u}_{\lambda,0}, \mathbf{u}_{\lambda,1})) \in \mathbb{Z}_q^{2m \times \lambda}$. Then $\mathcal{B}$ proceeds as follows:

Setup. It samples $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{v}_0, \mathbf{v}_1 \leftarrow \mathbb{Z}_q^n$. Then return $\mathsf{crs} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c)$.

Challenge. Upon receiving $(\mathsf{com}, L, \mathsf{st})$ from $\mathcal{A}_1$, where $\mathsf{com} = y_\epsilon$, it generates the challenge ciphertext for all $i \in [\lambda]$ as follows:

- Sample $t_{i,0}, t_{i,1} \leftarrow \mathbb{Z}_q$, $\mathbf{s}_{i,\kappa}, \ldots, \mathbf{s}_{i,\ell}, \mathbf{s}_{i,\ell+1} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{e}_{i,\kappa+1}, \ldots, \mathbf{e}_{i,\ell}, \mathbf{e}_{i,\ell+1} \leftarrow \mathcal{D}^{2m}_{\mathbb{Z},\sigma}$, $\mathbf{e}_{i,\ell+2} \leftarrow \mathcal{D}^m_{\mathbb{Z},\sigma}$.
- For all $j < \kappa$, it samples $\widetilde{\mathbf{c}}_{i,j} \leftarrow \mathbb{Z}_q^{2m}$ and set $\mathbf{c}_{i,j} = \widetilde{\mathbf{c}}_{i,j}$.
- For $j = \kappa$, it has two cases: if $j \le \ell$, it sets

$$\mathbf{c}_{i,j}^\top = -(\mathbf{u}_{i,0}^\top \mid \mathbf{u}_{i,1}^\top) + \mathbf{s}_{i,j}^\top \overbrace{(\overline{L}[j-1] \cdot \overline{\mathbf{G}} \mid L[j-1] \cdot \overline{\mathbf{G}})}^{\mathbf{T}_{j-1}}$$

  otherwise, it sets

$$\mathbf{c}_{i,j}^\top = -(\mathbf{u}_{i,0}^\top \mid \mathbf{u}_{i,1}^\top) + \mathbf{s}_{i,j}^\top \overbrace{(\overline{\mathbf{G}} \mid \overline{\mathbf{G}})}^{\mathbf{T}_\ell}$$

- For all $j > \kappa$, it generates $\mathbf{c}_{i,j}$ as usual.
- Sample $C_0, C_1 \leftarrow \{0,1\}^{\lambda \log c}$.

Finally, return $\mathsf{ct}_b = (\{t_{i,b}, \mathbf{c}_{i,j}\}_{i \in [\lambda], j \in [\ell+2]}, C_b)$ for $b \in \{0,1\}$.

Observe that for all $i \in [\lambda]$, we have

$$\begin{aligned}\mathbf{c}_{i,\kappa}^\top &= \boxed{-\mathbf{s}_{i,\kappa-1}^\top(\mathbf{A}_0 \mid \mathbf{A}_1) + \mathbf{e}_{i,\kappa}^\top} + \mathbf{s}_{i,\kappa}^\top \mathbf{T}_{\kappa-1} \\ &\approx_c -(\mathbf{u}_{i,0}^\top \mid \mathbf{u}_{i,1}^\top) + \mathbf{s}_{i,\kappa}^\top \mathbf{T}_{\kappa-1} \\ &\approx_s -(\mathbf{u}_{i,0}^\top \mid \mathbf{u}_{i,1}^\top) + \cancel{\mathbf{s}_{i,\kappa}^\top \mathbf{T}_{\kappa-1}}\end{aligned}$$

If all $(\mathbf{u}_{i,0}, \mathbf{u}_{i,1})$ are uniformly distributed over $\mathbb{Z}_q^{2m}$, then $\approx_s$ clearly holds and $\mathcal{B}$ hence simulates $\mathsf{G}_{3,\kappa}$; otherwise, $\mathcal{B}$ simulates $\mathsf{G}_{3,\kappa-1}$. Thus, this lemma is readily proved. □

Lemma 10. *Under matrix* $\mathsf{DLWE}_{n,(m+1),\lambda,q,\sigma_1}$ *assumption, it has* $\mathsf{G}_{3,\ell+1} \approx_c \mathsf{G}_4$.

Proof. Assume an algorithm $\mathcal{B}$ against matrix $\mathsf{DLWE}_{n,(m+1),\lambda,q,\sigma_1}$ assumption and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against the malicious OW security of WE. Initially, $\mathcal{B}$ is given a matrix DLWE instance $(\mathbf{A}, \mathbf{u})$ where $\mathbf{A} = (\mathbf{B} \mid \mathbf{d}) \in \mathbb{Z}_q^{n\times(m+1)}$ and $\mathbf{U} = ((\mathbf{u}_1, w_1) \mid \cdots \mid (\mathbf{u}_\lambda, w_\lambda)) \in \mathbb{Z}_q^{(m+1)\times\lambda}$. Then $\mathcal{B}$ proceeds as follows:

Setup. It samples $\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{v}_0 \leftarrow \mathbb{Z}_q^n$. Then set $\mathbf{v}_1 = \mathbf{v}_0 + \mathbf{d}$. Return $\mathsf{crs} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{B}, \mathbf{v}_0, \mathbf{v}_1, N, \ell, c)$. Since $\mathbf{d}$ is randomly sampled, $\mathcal{A}$ cannot distinguish the simulated crs from the real world.

Challenge. Upon receiving $(\mathsf{com}, L, \mathsf{st})$ where $\mathsf{com} = y_\epsilon$. For all $i \in [\lambda]$, it samples $t_{i,0}, t_{i,1} \leftarrow \mathbb{Z}_q$, $\mathbf{c}_{i,j} \leftarrow \mathbb{Z}_q^{2m}$ for all $j \in [\ell+1]$ and $C_0, C_1 \leftarrow \{0,1\}^{\lambda \log c}$. Set $\mathbf{c}_{i,\ell+2} = -\mathbf{u}_i$ and return $\mathsf{ct}_b = (\{t_{i,b}, \mathbf{c}_{i,j}\}_{i\in[\lambda], j\in[\ell+2]}, C_b)$ for $b \in \{0,1\}$.

Then $\mathcal{A}_1$ outputs a pair of messages $(\mathsf{m}_0, \mathsf{m}_1)$, then $\mathcal{B}$ can set $\mathsf{k}_0 = C_0 \oplus \mathsf{m}_0 = k_{1,0}||\cdots||k_{\lambda,0}$ and $\mathsf{k}_1 = C_1 \oplus \mathsf{m}_1 = k_{1,1}||\cdots||k_{\lambda,1}$. For all $i \in [\lambda]$, we define the following sets

$$S_{i,0} = \{x \in \mathbb{Z}_q \mid \lfloor x \rceil_c = k_{i,0}\}, \quad S_{i,1} = \{x \in \mathbb{Z}_q \mid \lfloor x \rceil_c = k_{i,1}\}$$
$$T_i = \{(x_0 - x_1) \in \mathbb{Z}_q \mid x_0 \in S_{i,0}, x_1 \in S_{i,1}\}$$

$\mathcal{B}$ will check whether $w_i + t_{i,0} - t_{i,1} \in T_i$ for all $i \in [\lambda]$. If all these pass, return 1; otherwise, return 0. Based on the distribution of the matrix DLWE instance, we analyze the probability that $\mathcal{B}$ outputs 1 as follows:

- If the matrix DLWE instance is uniformly distributed, i.e., $\mathbf{U} = ((\mathbf{u}_1, w_1) \mid \cdots \mid (\mathbf{u}_\lambda, w_\lambda))$ are uniform over $\mathbb{Z}_q^{(m+1)\times\lambda}$, it exactly simulates G_4. Then for each $i \in [\lambda]$, the probability that $w_i + t_{i,0} - t_{i,1} \in T_i$ is at most

$$\frac{|T_i|}{q} = \frac{(|S_{i,0}| + |S_{i,1}|)}{q} = \frac{2p}{q} = \frac{2}{c}$$

  By setting $c = 4$, the probability that $\mathcal{B}$ outputs 1 is at most $2^{-\lambda}$.
- If the matrix DLWE instance is structured, i.e., $\mathbf{U} = ((\mathbf{u}_1, w_1) \mid \cdots \mid (\mathbf{u}_\lambda, w_\lambda)) = ((\mathbf{s}_{1,\ell+1}^\top \mathbf{B}, \mathbf{s}_{1,\ell+1}^\top \mathbf{d}) \mid \cdots \mid (\mathbf{s}_{\lambda,\ell+1}^\top \mathbf{B}, \mathbf{s}_{\lambda,\ell+1}^\top \mathbf{d}))$ (we ignore the noisy terms in this case), it exactly simulates $\mathsf{G}_{3,\ell+1}$. If $\mathcal{A}_2$ can correctly recover $(\mathsf{m}_0, \mathsf{m}_1)$, then $(\mathsf{k}_0, \mathsf{k}_1)$ are also structured. For all $i \in [\lambda]$, it must have $-\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_1 + t_{i,0} \in S_{i,0}$ and $-\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_0 + t_{i,1} \in S_{i,1}$. Then it would have

$$\begin{aligned}&-\mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon + \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_1 + t_{i,0} + \mathbf{s}_{i,0}^\top \mathbf{y}_\epsilon - \mathbf{s}_{i,\ell+1}^\top \mathbf{v}_0 - t_{i,1} \\ &= \mathbf{s}_{i,\ell+1}^\top (\mathbf{v}_1 - \mathbf{v}_0) + t_{i,0} - t_{i,1} \\ &= \mathbf{s}_{i,\ell+1}^\top \mathbf{d} + t_{i,0} - t_{i,1} \in T_i\end{aligned}$$

  Similar to Lemma 7, the probability that $\mathcal{B}$ outputs 1 at least equals to the probability that $\mathcal{A}_2$ correctly recover $(\mathsf{m}_0, \mathsf{m}_1)$.

Above all, if the probability that $\mathcal{A}_2$ correctly recovers $(\mathsf{m}_0, \mathsf{m}_1)$ is non-negligible, $\mathcal{B}$ can also break the matrix $\mathsf{DLWE}_{n,(m+1),\lambda,q,\sigma_1}$ assumption with the same advantage. This proves $\mathsf{G}_{3,\ell+1} \approx_c \mathsf{G}_4$. □

Finally, we conclude the whole proof of malicious OW security. □

5 Malicious Laconic Oblivious Transfer

In this section, we construct malicious laconic OT from malicious witness encryption for equivocal vector commitment. The definition of malicious laconic OT is given in Section 5.1, then we give the construction in Section 5.2.

5.1 Definition

A malicious laconic OT consists of four algorithms as follows:

- $\mathsf{Setup}(1^\lambda, N) \to \mathsf{crs}$. It takes as input the security parameter 1^λ and the size of database N, then outputs a common reference string crs.
- $\mathsf{Hash}(\mathsf{crs}, D) \to (\mathsf{dig}, \mathsf{aux})$. It takes as input a common reference string crs and a database $D \in \{0,1\}^N$, then outputs a short digest dig and auxiliary information $\mathsf{aux} = (\mathsf{aux}_0, \ldots, \mathsf{aux}_{N-1})$.
- $\mathsf{Send}(\mathsf{crs}, \mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1) \to \mathsf{ct}$. It takes as input a common reference string crs, a digest dig, a location $L \in [0, N-1]$ and two messages $(\mathsf{m}_0, \mathsf{m}_1)$, then outputs a ciphertext ct.
- $\mathsf{Receive}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{ct}, L) \to \mathsf{m}$. With random access to aux, it takes as input a common reference string crs, a ciphertext ct and a location $L \in [0, N-1]$, then outputs a message m. This algorithm can also be rewritten as $\mathsf{Receive}(\mathsf{crs}, \mathsf{aux}_L, \mathsf{ct}, L)$.

Correctness. A laconic OT is δ-correct if for all $\lambda \in \mathbb{N}$, all databases $D \in \{0,1\}^N$, all memory locations L, and all messages $(\mathsf{m}_0, \mathsf{m}_1)$, it holds that

$$\Pr\left[\mathsf{m}_{D[L]} = \mathsf{Receive}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{ct}, L) \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{dig}, \mathsf{aux}) \leftarrow \mathsf{Hash}(\mathsf{crs}, D) \\ \mathsf{ct} \leftarrow \mathsf{Send}(\mathsf{crs}, \mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1) \end{array}\right] = 1 - \delta.$$

Analogously, it is also called perfectly correct if $\delta = 0$.

Efficiency. For all $\lambda \in \mathbb{N}$, all databases $D \in \{0,1\}^N$, it holds that $|\mathsf{dig}| = \mathsf{poly}(\lambda, \log N)$ and $|\mathsf{ct}| = \mathsf{poly}(\lambda, \log N)$. Besides, the runtime of Enc is bounded by $\mathsf{poly}(\lambda, \log N)$.

UC Security against Malicious Senders. We briefly describe the universal composability security [Can01] when facing a malicious sender: it also defines a context $\mathcal{Z}$ that oversees the execution of a protocol in one of two worlds. The "ideal world" execution involves a "dummy sender" (who is corrupted by an ideal simulator $\mathcal{S}$) interacting with a functionality F. The "real world" execution involves a real sender (who colludes with a real-world adversary $\mathcal{A}$) interacting only with an honest receiver in the laconic OT protocol. We refer to [Can01] for a detailed description of the executions, and a definition of the real-world ensemble $\mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}$ and the ideal-world ensemble $\mathsf{IDEAL}_{F,\mathcal{S},\mathcal{Z}}$. For all efficient adversaries $\mathcal{A}$ and all contexts $\mathcal{Z}$, a laconic OT protocol Π is said to be UC secure if there exists a simulator $\mathcal{S}$ such that

$$\mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} \approx_c \mathsf{IDEAL}_{F,\mathcal{S},\mathcal{Z}}.$$

Since our constructions are considered in the common reference string model, we define the common reference string functionality R_{G} as follows:

- Assume R runs with parties $P_1, \ldots, P_n$ and is initialized by an algorithm G;
- When receiving a message (sid, P_i, P_j) from P_i, let $\mathsf{crs} \leftarrow \mathsf{G}(1^n)$, send $(\mathsf{sid}, \mathsf{crs})$ to P_i and send (crs, P_i, P_j) to the adversary;
- When receiving a message (sid, P_i, P_j) from P_j (and only P_j), send $(\mathsf{sid}, \mathsf{crs})$ to P_j and to the adversary, and abort.

We can define the ideal functionality F of laconic OT as follows:

- Assume F runs with an ideal sender and an ideal receiver;
- When receiving $(\mathsf{sid}, \mathit{send}, L, \mathsf{m}_0, \mathsf{m}_1)$ from the sender, store $(L, \mathsf{m}_0, \mathsf{m}_1)$;
- When receiving $(\mathsf{sid}, \mathit{receive}, D)$ from the receiver, check if some inputs $(L, \mathsf{m}_0, \mathsf{m}_1)$ have been recorded for sid. If this is the case, send $(L, \mathsf{m}_{D[L]})$ to the receiver; otherwise, send $\perp$.

IND Security against Malicious Receivers. For all $\lambda, N \in \mathbb{N}$ and all efficient adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, it defines the following advantage

$$\mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A},\beta}(\lambda) = \left| \Pr\left[\alpha = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{crs}) \\ \mathsf{ct} \leftarrow \mathsf{Send}(\mathsf{crs}, \mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1) \\ \alpha \leftarrow \mathcal{A}_2(\mathsf{ct}, \mathsf{st}) \end{array}\right] - \Pr\left[\alpha = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{crs}) \\ \mathsf{ct} \leftarrow \mathsf{Send}(\mathsf{crs}, \mathsf{dig}, L, \ r_0, r_1\ ) \\ \alpha \leftarrow \mathcal{A}_2(\mathsf{ct}, \mathsf{st}) \end{array}\right] \right|$$

where $\beta \in \{0,1\}$, r_β is a random string and $r_{1-\beta} = \mathsf{m}_{1-\beta}$. The indistinguishable sender privacy requires

$$\Pr\left[\mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A},0}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)} \wedge \mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A},1}(\lambda) > \frac{1}{\mathsf{poly}(\lambda)}\right] = \mathsf{negl}(\lambda).$$

Context Security against Malicious Receivers. For all $\lambda, N \in \mathbb{N}$, all inverse polynomial $\delta = \delta(\lambda)$ and probabilistic Turing machine $\mathcal{Z} = (\mathcal{Z}_1, \mathcal{Z}_2)$ running in time $\mathsf{poly}(\lambda)$ (also called the context), there exists two simulation algorithms $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Send}})$ and an extraction algorithm Ext such that except for finitely many λ, we claim that

$$\mathsf{Adv}^{\mathsf{Ctxt}}_{\mathcal{Z}}(\lambda, \delta) = \left| \Pr\left[b' = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N) \\ (\mathsf{dig}, \mathsf{st}) \leftarrow \mathcal{Z}_1(\mathsf{crs}, r_1) \\ b' \leftarrow \mathcal{Z}_2^{\mathsf{Send}(\mathsf{crs}, \mathsf{dig}, \cdot, \cdot, \cdot)}(\mathsf{st}, r_2) \end{array}\right] - \Pr\left[b' = 1 \;\middle|\; \begin{array}{l} (\mathsf{crs}, \mathsf{td}) \leftarrow \widetilde{\mathsf{Setup}}(1^\lambda, N) \\ (\mathsf{dig}, \mathsf{st}) \leftarrow \mathcal{Z}_1(\mathsf{crs}, r_1) \\ (b, \mathsf{aux}) \leftarrow \mathsf{Ext}_{\mathcal{Z}}(\mathsf{crs}, \mathsf{st}, \mathsf{dig}, \delta, r^*) \\ b' \leftarrow \mathcal{Z}_2^{\widetilde{\mathsf{Send}}(\mathsf{td}, \mathsf{dig}, \mathsf{aux}, \cdot)}(\mathsf{st}, r_2) \end{array}\right] \right| < \delta.$$

where r_1, r_2 and r^* are random tapes. In the ideal world (underside), $\widetilde{\mathsf{Send}}$ only takes the output from ideal functionality F, and the overheads of $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{Send}})$ are independent of $\mathcal{Z}_1$ and $\mathcal{Z}_2$, where $\widetilde{\mathsf{Send}}$ just acquires the output of the ideal functionality F. In addition, Ext only depends on $\mathcal{Z}_2$. The above model is only considered in the single-session setting, but can be adapted into the multi-session setting via standard hybrid arguments.

Lemma 11 ([DGGM19]). *Assuming laconic OT satisfies the IND security against malicious receivers, then it also satisfies the context security against malicious receivers.*

5.2 Construction

Assume a witness encryption scheme $\mathsf{WE} = (\mathsf{Enc}, \mathsf{Dec})$ for an equivocal vector commitment $\mathsf{EVC} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{Open}, \mathsf{Ver}, \mathsf{TSetup}, \mathsf{TCom}, \mathsf{Equiv})$ over vector space $\{0,1\}^N$, our malicious laconic OT works as follows:

- $\mathsf{Setup}(1^\lambda, N)$: Run $\mathsf{crs}_{\mathsf{EVC}} \leftarrow \mathsf{EVC.Setup}(1^\lambda, N)$ and output $\mathsf{crs} = \mathsf{crs}_{\mathsf{EVC}}$.
- $\mathsf{Hash}(\mathsf{crs}, D)$: Run $(\mathsf{com}_{\mathsf{EVC}}, \mathsf{aux}_{\mathsf{EVC}}) \leftarrow \mathsf{EVC.Com}(\mathsf{crs}_{\mathsf{EVC}}, D)$ and $\pi_L \leftarrow \mathsf{EVC.Open}(\mathsf{crs}, \mathsf{com}, \mathsf{aux}, L)$ for all $L \in [0, N-1]$. Output $\mathsf{dig} = \mathsf{com}_{\mathsf{EVC}}$ and $\mathsf{aux} = (D, \pi_0, \ldots, \pi_{N-1})$.
- $\mathsf{Send}(\mathsf{crs}, \mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1)$: Sample randomness $r \leftarrow R$ and compute $\mathsf{ct}_0 \leftarrow \mathsf{WE.Enc}(\mathsf{crs}, (\mathsf{com}, L, 0), \mathsf{m}_0; r)$ and $\mathsf{ct}_1 \leftarrow \mathsf{WE.Enc}(\mathsf{crs}, (\mathsf{com}, L, 1), \mathsf{m}_1; r)$. Output $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$.
- $\mathsf{Receive}^{\mathsf{aux}}(\mathsf{crs}, \mathsf{ct}, L)$: Output $\mathsf{m}_{D[L]} \leftarrow \mathsf{WE.Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, D[L]), \pi_L)$.

Correctness and Efficiency. The correctness of our construction follows the underlying WE scheme. Observe that above construction almost inherits the efficiency of the underlying WE and EVC schemes, which satisfies the efficiency requirement of laconic OT.

Malicious Security. Next, we present the following theorems to claim the malicious security of above laconic OT scheme.

Theorem 9. *Assuming witness encryption satisfies the malicious IND security, then the above construction achieves the context security (against malicious receivers) as defined in Section 5.1.*

Proof. Now we prove the context security against malicious receivers for our laconic OT, assuming the malicious IND security of WE. We remark that the malicious IND security of WE indeed implies the IND security (against malicious receivers) of laconic OT. Assume an adversary $\mathcal{A}$ against the IND security of laconic OT, then we can construct an algorithm $\mathcal{B}$ against the malicious IND security of WE. $\mathcal{B}$ will roughly run as follows:

Initially, $\mathcal{B}$ invokes the challenger of WE and receives $\mathsf{crs}_{\mathsf{EVC}}$, then returns $\mathsf{crs} = \mathsf{crs}_{\mathsf{EVC}}$ to $\mathcal{A}$. Upon receiving $(\mathsf{dig}, L, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st})$ from $\mathcal{A}$, $\mathcal{B}$ directly forwards the same message to the challenger who will return the challenge ciphertexts $(\mathsf{ct}_0, \mathsf{ct}_1)$. Then $\mathcal{B}$ returns $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ to $\mathcal{A}$ and forward the response of $\mathcal{A}$ to the challenger, thus completing the reduction. Therefore, our laconic OT scheme is IND secure against malicious receivers.

By Lemma 11, for any laconic OT, the IND security is equivalent to the context security, so our construction equivalently achieves the context security. □

Theorem 10. *Assuming vector commitment satisfies equivocality, then the above construction achieves the UC security (against malicious senders) as defined in Section 5.1.*

Proof. Now we prove the UC security against malicious senders for our laconic OT, assuming the existence of equivocal vector commitment. Firstly, for the common reference string functionality R, since the crs output by EVC.Setup is statistically indistinguishable from the $\widetilde{\mathsf{crs}}$ output by EVC.TSetup. Therefore, by a standard hybrid argument, the execution of $R_{\mathsf{EVC.Setup}}$ is indistinguishable from the execution of $R_{\mathsf{EVC.TSetup}}$. In the following, we can replace the execution of $R_{\mathsf{EVC.Setup}}$ with that of $R_{\mathsf{EVC.TSetup}}$.

In the real world, the context $\mathcal{Z}$ initializes $\widetilde{\mathsf{crs}}_{\mathsf{EVC}} \leftarrow \mathsf{EVC.TSetup}(1^\lambda, N)$ (ignoring td), and set $\mathsf{crs} = \widetilde{\mathsf{crs}}_{\mathsf{EVC}}$. Note that $\mathcal{Z}$ is not aware of the difference between EVC.TSetup and EVC.Setup since only $\widetilde{\mathsf{crs}}$ is revealed to $\mathcal{Z}$. Then $\mathcal{Z}$ schedules an arbitrary number of sessions. In each session, $\mathcal{Z}$ chooses an input D for the honest receiver who computes $(\mathsf{com}_{\mathsf{EVC}}, \mathsf{aux}_{\mathsf{EVC}}) \leftarrow \mathsf{EVC.Com}(\mathsf{crs}, D)$. Next, $\mathcal{Z}$ produces arbitrary ct and the honest receiver will output $\mathsf{WE.Dec}(\mathsf{crs}, \mathsf{ct}, (\mathsf{com}, L, D[L]), \pi_L)$.

In the ideal world, $\mathcal{Z}$ initializes $(\widetilde{\mathsf{crs}}_{\mathsf{EVC}}, \mathsf{td}_{\mathsf{EVC}}) \leftarrow \mathsf{EVC.TSetup}(1^\lambda, N)$, and set $\mathsf{crs} = \widetilde{\mathsf{crs}}_{\mathsf{EVC}}$. Then $\mathcal{Z}$ schedules an arbitrary number of sessions. In each session, $\mathcal{Z}$ chooses an arbitrary D (not known to $\mathcal{S}$) for the honest receiver who will forward D to the ideal functionality F. The simulator $\mathcal{S}$ generates $(\widetilde{\mathsf{com}}_{\mathsf{EVC}}, \mathsf{st}_{\mathsf{EVC}}) \leftarrow \mathsf{EVC.TCom}(\widetilde{\mathsf{crs}}_{\mathsf{EVC}}, \mathsf{td})$ and return $\widetilde{\mathsf{com}}_{\mathsf{EVC}}$ to $\mathcal{Z}$, then $\mathcal{S}$ receives $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1)$ from $\mathcal{Z}$ (with explicitly specifying L). It can compute the opening proofs $\widetilde{\pi}_{L,b} \leftarrow \mathsf{EVC.Equiv}(\widetilde{\mathsf{crs}}, \mathsf{td}, \mathsf{st}, L, b)$ for each $b \in \{0,1\}$, thus recovering $\mathsf{m}_b \leftarrow \mathsf{WE.Dec}(\widetilde{\mathsf{crs}}, \mathsf{ct}_b, (\widetilde{\mathsf{com}}_{\mathsf{EVC}}, L, b), \widetilde{\pi}_{L,b})$. Along this way, it manages to forward $(L, \mathsf{m}_0, \mathsf{m}_1)$ to the ideal functionality F. Thus, we complete the proof. □

References

[ABD+21] Navid Alamati, Pedro Branco, Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, and Sihang Pu. Laconic private set intersection and applications. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part III*, volume 13044 of *LNCS*, pages 94–125. Springer, Cham, November 2021. 1, 2, 4

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996. 10, 12

[ALOS22] Diego F. Aranha, Chuanwei Lin, Claudio Orlandi, and Mark Simkin. Laconic private set-intersection from pairings. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 111–124. ACM Press, November 2022. 4

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012*, pages 326–349. ACM, January 2012. 2

[BCG+13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 90–108. Springer, Berlin, Heidelberg, August 2013. 2

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Berlin, Heidelberg, August 2001. 12, 13

[BG18] Jonathan Bootle and Jens Groth. Efficient batch zero-knowledge arguments for low degree polynomials. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 561–588. Springer, Cham, March 2018. 3

[BGJP25] James Bartusek, Sanjam Garg, Abhishek Jain, and Guru-Vamsi Policharla. Laconic PSI on authenticated inputs and applications. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part V*, volume 16249 of *LNCS*, pages 262–293. Springer, Singapore, December 2025. 4

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Berlin, Heidelberg, August 2005. 12

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Berlin, Heidelberg, April 2012. 9, 12

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145. IEEE Computer Society Press, October 2001. 3, 5, 28

[CDG+17] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 33–65. Springer, Cham, August 2017. 1, 2, 4

[CF13] Dario Catalano and Dario Fiore. Vector commitments and their applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 55–72. Springer, Berlin, Heidelberg, February / March 2013. 8, 19

[CFK24] Matteo Campanelli, Dario Fiore, and Hamidreza Khoshakhlagh. Witness encryption for succinct functional commitments and applications. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14602 of *LNCS*, pages 132–167. Springer, Cham, April 2024. 5, 6

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 17–33. Springer, Berlin, Heidelberg, February 2005. 7, 15, 16

[DGGM19] Nico Döttling, Sanjam Garg, Vipul Goyal, and Giulio Malavolta. Laconic conditional disclosure of secrets and applications. In David Zuckerman, editor, *60th FOCS*, pages 661–685. IEEE Computer Society Press, November 2019. 2, 3, 4, 5, 7, 29

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 768–797. Springer, Cham, May 2020. 15

[DGI+19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 3–32. Springer, Cham, August 2019. 2

[DGM23] Nico Döttling, Phillip Gajland, and Giulio Malavolta. Laconic function evaluation for Turing machines. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 606–634. Springer, Cham, May 2023. 3

[DHM+24] Fangqi Dong, Zihan Hao, Ethan Mook, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and ABE for RAMs from (ring-)LWE. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 107–142. Springer, Cham, August 2024. 3

[DHMW24] Fangqi Dong, Zihan Hao, Ethan Mook, and Daniel Wichs. Laconic function evaluation, functional encryption and obfuscation for RAMs with sublinear computation. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 190–218. Springer, Cham, May 2024. 2, 3

[DKL+23] Nico Döttling, Dimitris Kolonelos, Russell W. F. Lai, Chuanwei Lin, Giulio Malavolta, and Ahmadreza Rahimi. Efficient laconic cryptography from learning with errors. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 417–446. Springer, Cham, April 2023. 2, 4, 8

[EGG+22] Harry Eldridge, Aarushi Goel, Matthew Green, Abhishek Jain, and Maximilian Zinkus. One-time programs from commodity hardware. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part III*, volume 13749 of *LNCS*, pages 121–150. Springer, Cham, November 2022. 2

[FHAS24] Nils Fleischhacker, Mathias Hall-Andersen, and Mark Simkin. Extractable witness encryption for KZG commitments and efficient laconic OT. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part II*, volume 15485 of *LNCS*, pages 423–453. Springer, Singapore, December 2024. 2, 4, 5, 6

[FKdP23] Dario Fiore, Dimitris Kolonelos, and Paola de Perthuis. Cuckoo commitments: Registration-based encryption and key-value map commitments for large spaces. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 166–200. Springer, Singapore, December 2023. 2, 3, 4, 5, 6

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Cham, August 2018. 4

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990. 2

[Gen10] Craig Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010. 3

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 467–476, 2013. 2, 6

[GHK+25] Sanjam Garg, Mohammad Hajiabadi, Dimitris Kolonelos, Abhiram Kothapalli, and Guru-Vamsi Policharla. A framework for witness encryption from linearly verifiable SNARKs and applications. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 504–539. Springer, Cham, August 2025. 5

[GHMR18] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. Registration-based encryption: Removing private-key generator from IBE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 689–718. Springer, Cham, November 2018. 2

[GJV23] Matthew Green, Abhishek Jain, and Gijs Van Laer. Efficient set membership encryption and applications. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 1080–1092. ACM Press, November 2023. 2, 4

[GKMR23] Noemi Glaeser, Dimitris Kolonelos, Giulio Malavolta, and Ahmadreza Rahimi. Efficient registration-based encryption. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 1065–1079. ACM Press, November 2023. 2, 3, 4

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989. 7, 16

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 10, 11, 12, 24

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Berlin, Heidelberg, December 2010. 2

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Berlin, Heidelberg, April 2008. 2

[GS17] Sanjam Garg and Akshayaram Srinivasan. Garbled protocols and two-round MPC from bilinear maps. In Chris Umans, editor, *58th FOCS*, pages 588–599. IEEE Computer Society Press, October 2017. 2

[GS18a] Sanjam Garg and Akshayaram Srinivasan. Adaptively secure garbling with near optimal online complexity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 535–565. Springer, Cham, April / May 2018. 2

[GS18b] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 468–499. Springer, Cham, April / May 2018. 2

[GVW20] Rishab Goyal, Satyanarayana Vusirikala, and Brent Waters. New constructions of hinting PRGs, OWFs with encryption, and more. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 527–558. Springer, Cham, August 2020. 2, 4

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. 2

[HLL23] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices. In *64th FOCS*, pages 415–434. IEEE Computer Society Press, November 2023. 4

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 169–186. Springer, Berlin, Heidelberg, May 2007. 16

[HWW25] Yao-Ching Hsieh, Brent Waters, and David J. Wu. A generic approach to adaptively-secure broadcast encryption in the plain model. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 336–365. Springer, Cham, May 2025. 16

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189. Springer, Cham, August 2017. 4, 5

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988. 1

[KNTY19] Fuyuki Kitagawa, Ryo Nishimaki, Keisuke Tanaka, and Takashi Yamakawa. Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. Springer, Cham, August 2019. 2

[LGW24] Ruiyang Li, Chun Guo, and Xiao Wang. Towards optimal garbled circuits in the standard model. Cryptology ePrint Archive, Report 2024/1907, 2024. 3

[LL93] Arjen K Lenstra and Hendrik W Lenstra. *The development of the number field sieve*, volume 1554. Springer Science & Business Media, 1993. 3

[LWYY25] Hanlin Liu, Xiao Wang, Kang Yang, and Yu Yu. BitGC: Garbled circuits with 1 bit per gate. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VII*, volume 15607 of *LNCS*, pages 437–466. Springer, Cham, May 2025. 3

[LY10] Benoît Libert and Moti Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 499–517. Springer, Berlin, Heidelberg, February 2010. 7, 17

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994. 2

[Mic18] Daniele Micciancio. On the hardness of learning with errors with binary secrets. *Theory of Computing*, 14(1):1–17, 2018. 11

[MORS26] Pierre Meyer, Claudio Orlandi, Lawrence Roy, and Peter Scholl. Computationally succinct authentication from dcr: attribute-based laconic function evaluation and more. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 314–344. Springer, 2026. 4

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012. 8, 10, 11, 12

[QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th FOCS*, pages 859–870. IEEE Computer Society Press, October 2018. 3

[RAB81] MO RABIN. How to exchange secrets by oblivious transfer. *Technical Memo TR-81*, 1981. 1

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. 9, 11

[Ros22] Razvan Rosie. Adaptively secure laconic function evaluation for NC^1. In Steven D. Galbraith, editor, *CT-RSA 2022*, volume 13161 of *LNCS*, pages 427–450. Springer, Cham, March 2022. 2

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997. 4

[Wee24] Hoeteck Wee. Circuit ABE with poly(depth, λ)-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024. 4

[Wee25] Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from succinct LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 34–62. Springer, Cham, May 2025. 4

[WW23] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 385–416. Springer, Cham, April 2023. 7, 13