Fine-Grained Cryptography Revisited

Shohei Egashira[1], Yuyu Wang[2], and Keisuke Tanaka[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
egashira.s.aa@m.titech.ac.jp, keisuke@is.titech.ac.jp
[2] University of Electronic Science and Technology of China, Chengdu, China
wangyuyu@uestc.edu.cn

Abstract. *Fine-grained cryptographic primitives* are secure against adversaries with bounded resources and can be computed by honest users with less resources than the adversaries. In this paper, we revisit the results by Degwekar, Vaikuntanathan, and Vasudevan in Crypto 2016 on fine-grained cryptography and show constructions of three key fundamental fine-grained cryptographic primitives: *one-way permutations*, *hash proof systems* (which in turn implies a *public-key encryption scheme against chosen chiphertext attacks*), and *trapdoor one-way functions*. All of our constructions are computable in NC^1 and secure against (*non-uniform*) NC^1 circuits under the widely believed worst-case assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$.

Keywords: Fine-grained cryptography · NC^1 circuit · One-way permutation · Hash proof system · Trapdoor one-way function

1 Introduction

1.1 Background

To prove the security of a cryptographic scheme, we typically reduce the security to some computational hardness assumption with a precise security definition. Due to the fact that most assumptions are unproven, it is desirable to make the underlying assumptions as weak as possible. However, it turns out to be very hard to construct a public-key cryptographic scheme without assuming the existence of one-way functions (OWF). Moreover, for a vast majority of primitives (including public-key encryption (PKE)), we further need to assume the hardness of specific problems such as factoring, discrete-logarithm, learning with errors, etc.. It still remains open whether it is possible to construct even basic cryptographic primitives under no assumptions, or at least mild complexity-theoretic assumptions. For instance, the complexity-theoretic assumption $\mathsf{NP} \not\subseteq \mathsf{BPP}$, which is strictly weaker than the assumption of OWFs, has been proven to be insufficient for constructing even OWFs as shown by Akavia et al. [4].

Due to the difficulty of directly constructing cryptographic primitives against any polynomial probabilistic time adversaries based on mild complexity-theoretic assumptions such as $\mathsf{NP} \not\subseteq \mathsf{BPP}$, a line of beautiful works focused on fine-grained cryptographic primitives [17], where (1) the resource of an adversary is a-prior bounded, (2) an honest party can run the algorithms with less resource than an adversary, and (3) the underlying assumption is extremely mild.

Merkle [37] initialized the study in this field by constructing a non-interactive key exchange scheme, which can be run in time $O(n)$ and adversaries running in time $o(n^2)$ cannot break the security. The construction only requires random functions (i.e., the random oracle). Subsequent to his work, Biham et al. [10] showed the existence of strong OWFs based on the same assumption.

While Merkle restricted adversaries in the term of running time, Maurer considered a model where adversaries have infinite computing power but only restricted storage [36]. Afterwards, he proposed a key exchange protocol in this model [38]. Following these works, Cachin and Maurer [13] constructed a symmetric-key encryption scheme and a key exchange protocol which can be run with storage $O(s)$ and are unconditionally secure against adversaries with storage $o(s^2)$. Besides, there have been many other works focusing on primitives in this model [9, 44, 19, 8, 20, 21].

In the constant depth circuit model, Ajtai and Wigderson [3] constructed an unconditional secure pseudo-random generator. Then, Boppana and Lagarias [12] exploited the results by Ajitai [2] and Furst et al. [23], which shows that parity cannot be computed in size-bounded circuits, to achieve OWFs. The proposed OWF can be computed in AC^0 (constant-depth polynomial-sized) circuits consisting of AND, OR, and NOT gates of unbounded fan-in, while the inverse cannot. Afterwards, several works treating the same model have been proposed [30, 6, 45, 46].

Recently, Degwekar et al. [17] proposed fine-grained cryptographic primitives against adversaries captured by two (non-uniform) classes of adversaries, which are AC^0 and NC^1 (logarithmic-depth polynomial-sized) circuits consisting of AND, OR, and NOT gates of fan-in 2. They first constructed an unconditionally secure pseudorandom generator with arbitrary polynomial stretch, a weak pseudorandom function, and a secret-key encryption scheme, all of which are computable in AC^0 and secure against adversaries that are AC^0 circuits. Then, under the widely believed separation assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, they constructed a OWF, a pseudorandom generator, a collision-resistant hash function, and a semantically secure PKE scheme that are computable in NC^1 and secure against NC^1 circuits.

Following the above work, Campanelli and Gennaro [14] constructed a somewhat homomorphic encryption and a verifiable computation against NC^1 circuits. As in [17], the underlying assumption is $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$.

While the above sequence of works have achieved amazing success, it still remains open whether it is possible to construct other fine-grained primitives, such as one-way permutation (OWP), PKE against chosen ciphertext attacks (CCA), and even trapdoor one-way function (TDF).

1.2 Our Results and Techniques

In this paper, we propose several fine-grained cryptographic primitives under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. Specifically, we propose a OWP, a hash proof system (HPS) (which in turn derives a CCA-secure PKE scheme), and a TDF. All of them are computable in NC^1 and secure against adversaries captured by the class of NC^1 circuits. Since a lot of results have been devoted to constructing advanced primitives from these fundamental ones, our results greatly alleviate the efforts to achieve more fine-grained primitives from scratch.

Our constructions rely on the fact shown in the papers by Appelebaum, Ishai, and Kushilevitz [31, 5], that if $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$, there exist a distribution D_0^n over $n \times n$ matrices of rank $(n-1)$ and a distribution D_1^n over $n \times n$ matrices of rank n, which are indistinguishable for NC^1 circuits.

One-way permutation. As one of the most fundamental cryptographic primitives, OWP has been shown to be sufficient for constructing many primitives (e.g. pseudorandom generators [11] and universal one-way hash functions [39]). Compared with primitives built from OWFs which are not bijective (e.g., [42, 28]), ones built from OWPs are usually more efficient [7, 35].

In the previous work, Degwekar et al. [17] showed a construction of fine-grained OWFs in NC^1. Their construction relies on a randomized encoding of a boolean function f, which is a randomized function outputting the distribution related only to $f(x)$. Specifically, let $\hat{f} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{m+1} \in \mathsf{NC}^1$ be the randomized encoding of $f \in \oplus\mathsf{L}/\mathsf{poly}$, where the existence of $\hat{f}$ is shown in [5]. Then, their construction of a OWF is $g(x) = \hat{f}(0^n, x)$.[3] However, the domain and range of g are $\{0,1\}^m$ and $\{0,1\}^{m+1}$ respectively, i.e., the domain and range of g are inconsistent. Thus their construction is not a permutation. Moreover, since they define OWFs using randomized encoding directly, it is difficult to make their construction a permutation, i.e., it is not clear how to further achieve OWPs under the same worst-case assumption.

In this work, we propose a collection of OWPs and extend it to a OWP, both of which are computable in NC^1 and secure against NC^1 circuits under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$.

To achieve the goal, we exploit the two distributions D_0^n and D_1^n described above. Essentially, our idea is to construct a "lossy function family" $\{f_{\mathbf{M}}(\mathbf{x}) = \mathbf{M}\mathbf{x}\}_{\mathbf{M} \in D_1^n}$. We let $\mathbf{M} \leftarrow D_1^n$ and $\mathbf{M} \leftarrow D_0^n$ in the injective and lossy model respectively, and the indistinguishability between the two models can be reduced to the indistinguishability between D_1^n and D_0^n. Then we follow the Peikert-Waters [40] approach to prove that $f_{\mathbf{M}}$ in the injective model satisfies one-wayness. Furthermore, since a matrix $\mathbf{M} \leftarrow D_1^n$ is of full rank, it holds that $f_{\mathbf{M}}$ in the injective model is a permutation. Therefore, $\{f_{\mathbf{M}}(\mathbf{x}) = \mathbf{M}\mathbf{x}\}_{\mathbf{M} \in D_1^n}$ is a collection of OWPs. Next, we extend it to a OWP, i.e., we give a construction of OWP based on a collection of OWPs which satisfies the distribution of index sample algorithm is identical to the uniform distribution over index set as follows. For a collection of OWPs $\{f_i : D_i \to D_i\}_{i \in I}$ where I is an index set, define a function g with the domain $D := \bigcup_{i \in I}(\{i\} \times D_i)$ and $g((i, x) \in D) = (i, f_i(x))$.

[3] The one-wayness of g is based on the indistinguishability of the output distributions of $\hat{f}$ conditioned on $f(x) = 0$ and $f(x) = 1$, which can be reduced to $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$.

Since f_i is a permutation and one-way, g is a permutation and one-way as well, i.e., g is a OWP.

Hash proof system and CCA secure PKE scheme. The notion of HPS, which can be treated as designated verifier non-interactive zero-knowledge proof system for a language, was first introduced by Cramer and Shoup [16] for the purpose of constructing a CCA secure PKE scheme. An HPS allows one to generate a valid proof π proving that a statement x is in a language L by using a witness w and a public key pk. Also, one can generate a valid proof for x (not necessarily in L) by using only a secret key sk. For $x \in L$, proofs generated in these two ways should be the same. Typically, L is required to be a hard subset membership one, i.e., statements sampled from inside and outside the language should be indistinguishable. Furthermore, an HPS usually satisfies universality and smoothness. Universality means that for fixed x outside L and pk, the entropy of π is high enough (due to the entropy of sk). Smoothness means that for x outside L, the distribution of π honestly generated with sk is close to the uniform distribution in the proof space. HPSs are very versatile. Besides the application of PKE schemes, they play important roles in constructing various primitives, such as password authenticated key exchange [26, 34], oblivious transfer [33, 1], and zero-knowledge arguments [32].

In previous works, there has been no known way to construct HPSs that is computable in NC^1 and secure against adversaries bounded in NC^1 yet. Note that HPS is a quite different primitive from the ones in [17, 14], and its instantiation cannot be achieved via some simple extension. The main bottleneck is that it is not clear how to construct an HPS, where we can reduce the hardness of the subset membership problem to the indistinguishability between D_0^n and D_1^n. To overcome this problem, we define two sets L and L' that are identical to the supported language in a somewhat sophisticated way. The interesting part is that we can reduce the indistinguishability between L' and X/L to that between D_0^n and D_1^n. Also we did very careful analysis on the entropy of secret keys with respect to fixed public keys the to prove smoothness and universality. More details are given as follows.

In this work, we propose the first HPS that is computable in NC^1 and secure against NC^1 adversaries based on the worst-case assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$.

Our idea is to let a proof in the HPS be of the form $\mathbf{sk}^\top \mathbf{M}^\top \mathbf{w}$, where $\mathbf{M} \leftarrow D_0^n$, $\mathbf{x} = \mathbf{M}^\top \mathbf{w}$ is the statement with witness $\mathbf{w}$, $\mathbf{sk}$ is the secret key, and $\mathbf{M}\,\mathbf{sk} = \mathbf{pk}$ is the public key. A proof can be generated as either $\mathbf{pk}^\top \mathbf{w}$ or $\mathbf{sk}^\top \mathbf{x}$. The language that our HPS supports is $\mathrm{Im}(\mathbf{M}^\top)$. To achieve the hardness of our subset membership problem, we exploit the fact that $\mathrm{Im}(\mathbf{M}^\top)$ is identical to

$$L = \{\mathbf{x}|\mathbf{w} \in 1 \times \{0,1\}^{n-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\} = \{\mathbf{x}|\mathbf{w} \in 0 \times \{0,1\}^{n-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\},$$

which is indistinguishable from

$$\begin{aligned} L' &= \{\mathbf{x}|\mathbf{w} \in 0 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\} \\ &= \{\mathbf{x}|\mathbf{w} \in 1 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\}, \end{aligned}$$

where we sample $\mathbf{M}$ as $\mathbf{M} \leftarrow D_1^n$ instead of $\mathbf{M} \leftarrow D_0^n$ for L'. Then, we can reduce the indistinguishability between the uniform distributions over L' and $X \setminus L$ to that between D_0^n and D_1^n. To prove the universality and smoothness, we show that for one **pk**, there exist different valid secret keys, which lead to different outputs for any statement not in the language. Hence, the entropy of the proof is high due to the entropy of the secret key for a fixed **pk** and statement. We refer the reader to Section 4 for further details.

The proof size of the above scheme is only one single bit, while we can extend it to an HPS with multi-bit proofs by running many HPSs in parallel and show that the extension is still computable in NC^1 and secure against NC^1 circuits.

We now can instantiate the generic constructions [16] of a CCA-secure PKE scheme with our HPSs. The resulting scheme is secure against NC^1 circuits allowed to make constant rounds of adaptive decryption queries, while in each round, it can make arbitrary polynomial number of queries. This restriction is natural and defined in the same way as the adversaries for the NC^1-verifiable computation scheme in [14].

As far as we know, this is the first PKE that is CCA secure against NC^1 circuits under a mild complexity-theoretic assumptions, and there is no known way to make the PKE in [17] and the somewhat homomorphic encryption scheme in [14], which are malleable, CCA secure.

Trapdoor one-way function. TDF is a fundamental primitive introduced by Diffie and Hellman [18]. Unlike PKE schemes, where the decryption algorithm only recovers the plaintext (not including the internal randomness used in the encryption procedure), the inversion algorithm of a TDF recovers the entire preimage. The property of TDF mentioned above is useful in many applications, where proofs of well-formedness are required [24]. However, in the same time, it makes constructing TDFs very challenging.

In the previous works [17, 14], the PKEs use randomness in the encrypting procedures and it is difficult to recover the randomness in the decrypting procedures since the constructions recover the plaintexts by canceling the randomness using the property of the kernel of $\mathbf{M} \leftarrow D_0^n$. Namely, it is not easy to extend their construction to achieve a TDF. In fact, it has been shown that a TDF cannot be built from a PKE scheme in a black-box way [27][4]. On the other hand, it seems that there is a naive approach to construct a TDF f by defining it in the same way as our OWP, i.e., $f(\mathbf{x}) = \mathbf{M}_1\mathbf{x}$ where $\mathbf{M}_1$ is a sampled from D_1^n, and sample the inverse $\mathbf{M}_1^{-1}$ or some other elements that can be used to solve linear equations efficiently as the trapdoor. However, there is no known way to perform such a sampling procedure in NC^1 circuits. Therefore, some more sophisticated approach has to be taken.

In this work, we propose a TDF computable in NC^1 and secure against NC^1 circuits based on $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. The intuition is as follows.

[4] There is no rigorous proof showing that the separation holds for NC^1, while it is an evidence that TDF is not easy to achieve.

We first change the domain to $\{0,1\}^t \times (L \times X \setminus L)^t$ where $\mathbf{M} \leftarrow D_0^n$, $L = \text{Im}(\mathbf{M})$, and $X = \{0,1\}^n$. On input $(x, (\mathbf{c}_1, \mathbf{c}_1'), \cdots, (\mathbf{c}_t, \mathbf{c}_t')) \in \{0,1\}^t \times (L \times X \setminus L)^t$, our TDF computes $y = f(\mathbf{x})$, and additionally outputs $(\mathbf{c}_i, \mathbf{c}_i')$ if $x_i = 0$ and $(\mathbf{c}_i', \mathbf{c}_i)$ otherwise for all i. Here, f is a OWF that is computable in NC^1 and secure against NC^1 and x_i denotes the ith bit of x. Then, if we have a non-zero vector $\mathbf{k}$ in the kernel of $\mathbf{M}$, which is samplable in NC^1 [17], we can determine whether $\mathbf{x} \in \{0,1\}^n$ is in $\text{Im}(\mathbf{M}^\top)$ or $\{0,1\}^n \setminus \text{Im}(\mathbf{M}^\top)$ and recover x_i by checking whether $\mathbf{c}_i$ and $\mathbf{c}_i'$ are swapped. This provides us an efficiently samplable trapdoor. Due to the subset membership problem for $L = \text{Im}(\mathbf{M})$ we described before, the uniform distributions over $\text{Im}(\mathbf{M}^\top)$ and $\{0,1\}^n \setminus \text{Im}(\mathbf{M}^\top)$ are indistinguishable when $\mathbf{M}$ is a matrix sampled from D_0^n. Therefore, the adversary in the one-wayness game can only obtain information on $f(x)$ (which is one-way) and the additional pairs do little help to it.

The above technique of sampling additional pairs is called *bits planting* which was used by Garg et al. [25] to construct a TDF based on the computational Diffie-Hellman problem. Although both our construction and the one in [25] aim at constructing trapdoor TDFs, we use the *bits planting* in a different way. In [25], this technique is exploited to recover the randomness used in the computation procedure of the TDF (see [25] for details), while in our work, we use it to avoid sampling the inverse of $\mathbf{M}$ so that every operation can be performed in NC^1.

1.3 Possibility on the Extension from Our Proposed NC^1 Fine-Grained Primitives

As described above, the fundamental cryptographic primitives we considered play key roles in a great deal of applications. Hence, our results directly imply the existence of more advanced NC^1-fine-grained primitives. As a simple instance, besides CCA secure PKE schemes, our HPS immediately implies the existence of a non-interactive key exchange scheme according to the recent construction by [29]. However, some NC^1 primitives can not be directly derived from existing ones by adopting previous generic conversions in the polynomial-time world since the resulting primitive may not be in NC^1 any more. For example, although it is well known that pseudorandom functions can be constructed from OWF/OWPs, ones in NC^1 are neither implied by our NC^1-OWP nor the OWF in [17]. It remains open how to construct such fine-grained primitives, and we believe that our works will serve a good starting point.

1.4 Comparison with the Proceedings Version

This is the full version of the paper that appeared at ASIACRYPT 2019 [22]. In this full version, we give the transformation from collections of OWPs to OWPs. We also give the extension to a multi-bit perfectly smooth and strong universal$_1$ NC^1-HPS against NC^1 and a further extension to a strong universal$_2$ NC^1-HPS against NC^1, where the language supports $\{0,1\}^n$. Based on the resulting HPS, we give an NC^1-CCA-secure PKE scheme against NC^1. These results are respectively given in Sections A, B, C, and D in Supplementary Material.

We also note that in the previous version, there is a subtle flaw in the proof for the subset membership problem of our HPS. Roughly, the matrix $\mathbf{M}$ included in the public parameter is sampled from $\mathsf{OneSamp}(\lambda)$ in the final game, while it should be sampled from $\mathsf{ZeroSamp}(\lambda)$ in the original definition. In this version, we fix the proof by exploiting the underlying assumption twice when switching the distribution of $\mathbf{M}$. Compared with the previous version, the proof is more technical, while the construction remains the same. We refer the reader to Section 4 for details.

2 Preliminaries

2.1 Notation

For a distribution D, we denote sampling x according to D by $x \leftarrow D$. For a set S, we denote sampling x from S uniformly at random by $x \leftarrow S$. We denote the set $\{1, \cdots, n\}$ by $[n]$ and the ith element of a vector $\boldsymbol{x}$ by x_i. For a vector $\mathbf{x} \in \{0,1\}^*$, $\mathbf{x}$ will be regarded by default as a column vector. For a matrix $\mathbf{M}$, we denote the sets $\{\mathbf{y} \mid \exists \mathbf{x} \;\; s.t. \;\; \mathbf{y} = \mathbf{M}\mathbf{x}\}$ and $\{\mathbf{x} \mid \mathbf{M}\mathbf{x} = 0\}$ by $\mathrm{Im}(\mathbf{M})$ and $\mathrm{Ker}(\mathbf{M})$ respectively. Let X and Y be random variables over a finite set S. The *statistical distance* between X and Y is defined to be

$$\mathrm{Dist}(X, Y) = \frac{1}{2} \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|.$$

We say that X and Y are ϵ-close if $\mathrm{Dist}(X, Y) \leq \epsilon$

We note that all arithmetic computations are over $GF(2)$ in this work. Namely, all arithmetic computations are performed with a modulus of 2. By negl we denote an unspecified negligible function.

2.2 Definitions

In this section, we recall the definitions of a function family, NC^1 circuits, and $\oplus\mathsf{L/poly}$.

Definition 1 (Function Family) *A function family is a family of (possibly randomized) functions $\mathcal{F} = \{f_\lambda\}_{\lambda \in \mathbb{N}}$, where for each λ, f_λ has a domain D^f_λ and a range R^f_λ.*

Definition 2 (NC^1) *The class of (non-uniform) NC^1 function families is the set of all function families $\mathcal{F} = \{f_\lambda\}$ for which there is a polynomial p and constant c such that for each λ, f_λ can be computed by a (randomized) circuit of size $p(\lambda)$, depth $c\log(\lambda)$ and fan-in 2 using* AND*,* OR*, and* NOT *gates.*

Definition 3 ($\oplus\mathsf{L/poly}$) $\oplus\mathsf{L/poly}$ *is the set of all boolean function families $\mathcal{F} = \{f_\lambda\}$ for which there is a constant c such that for each λ, there is a non-deterministic Turing machine M_λ such that for each input x with length λ, $M_\lambda(x)$ uses at most $c\log(\lambda)$ space, and $f_\lambda(x)$ is equal to the parity of the number of accepting paths of $M_\lambda(x)$.*

We now give the lemma about the number of solutions for the linear equations defined by a matrix. It is straightforwardly follows from the fact that the rank of $\mathcal{A}$ is $n-1$.

Lemma 1 *For any $n \times n$ matrix $\mathbf{A}$, if the rank of $\mathbf{A}$ is $n-1$ and all arithmetic computations are over $GF(2)$, then for any $\mathbf{y} \in \mathrm{Im}(\mathbf{A})$, there exist and only exist two different vectors $\mathbf{x}$ and $\mathbf{x}'$ such that $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' = \mathbf{y}$.*

2.3 Definitions in Fine-Grained Cryptography

In this section, we define several cryptographic primitives which are secure against restricted complexity classes of adversaries and easy to run for honest parties. In the following definitions, we denote the class of honest parties by $\mathcal{C}_1$ i.e., function families that compose the primitive are in the class $\mathcal{C}_1$ and the class of adversaries by $\mathcal{C}_2$, and the condition $\mathcal{C}_1 \subseteq \mathcal{C}_2$ is implicit in each definition and hence left unmentioned.

Definition 4 (One-Way Function [17]) *Let l be a polynomial in λ. Let $\mathcal{F} = \{f_\lambda : \{0,1\}^\lambda \to \{0,1\}^{l(\lambda)}\}$ be a function family. $\mathcal{F}$ is a $\mathcal{C}_1$-*one-way function (OWF) *against $\mathcal{C}_2$ if:*

- **Computability:** *For each λ, f_λ is deterministic.*
- **One-wayness:** *For any $\mathcal{G} = \{g_\lambda : \{0,1\}^{l(\lambda)} \to \{0,1\}^\lambda\}$ and any $\lambda \in \mathbb{N}$:*

$$\Pr\left[f_\lambda(g_\lambda(y)) = y \,\middle|\, \begin{array}{c} x \leftarrow \{0,1\}^\lambda \\ y = f_\lambda(x) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

Definition 5 (One-Way Permutation) *Let $\mathcal{F} = \{f_\lambda : D_\lambda \to D_\lambda\}$ be a function family. $\mathcal{F}$ is a $\mathcal{C}_1$-*one-way permutation (OWP) *against $\mathcal{C}_2$ if:*

- **Permutation:** *For each λ, f_λ is a permutation.*
- **One-wayness:** *For any $\mathcal{G} = \{g_\lambda : D_\lambda \to D_\lambda\}$ and any $\lambda \in \mathbb{N}$:*

$$\Pr\left[g_\lambda(y) = x \,\middle|\, \begin{array}{c} x \leftarrow D_\lambda \\ y = f_\lambda(x) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

Definition 6 (Collection of OWPs) *Let $\mathcal{KeyGen} = \{\mathsf{KeyGen}_\lambda : \phi \to K_\lambda\}$ and $\mathcal{E}val = \{\mathsf{Eval}_\lambda : K_\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda\}$ be function families. $(\mathcal{KeyGen}, \mathcal{E}val)$ is a collection of $\mathcal{C}_1$-OWPs against $\mathcal{C}_2$ if:*

- **Permutation:** *For each λ and $k \leftarrow \mathsf{KeyGen}_\lambda$, $\mathsf{Eval}_\lambda(k,\cdot) : D_{\lambda,k} \to D_{\lambda,k}$ is a permutation where $D_{\lambda,k} \subseteq \{0,1\}^\lambda$.*
- **One-wayness:** *For any $\mathcal{G} = \{g_\lambda : K_\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda\}$ and any $\lambda \in \mathbb{N}$:*

$$\Pr\left[g_\lambda(k,y) = x \,\middle|\, \begin{array}{c} k \leftarrow \mathsf{KeyGen}_\lambda \\ x \leftarrow D_{\lambda,k} \subseteq \{0,1\}^\lambda \\ y = \mathsf{Eval}_\lambda(k,x) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

Definition 7 (Hash Proof System) *Let* $PP_\lambda = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, \mathit{\Pi}_\lambda, H_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$ *where* X_λ *is a finite non-empty set,* L_λ *is a subset of* X *such that* $x \in L_\lambda$ *iff there exists a witness* $w \in W_\lambda$ *with* $(x, w) \in R_\lambda \subset X_\lambda \times W_\lambda$, SK_λ *is a secret key space,* PK_λ *is a public key space,* $\mathit{\Pi}_\lambda$ *is a proof space,* $H_\lambda : SK_\lambda \times X_\lambda \to \mathit{\Pi}_\lambda$ *is a hash function,* $\alpha_\lambda : SK_\lambda \to PK_\lambda$ *is a projective map, and* aux_λ *is an auxiliary information. Define the following function families.*

- $\mathcal{S}etup = \{\mathsf{Setup}_\lambda : \phi \to PP_\lambda\}$ *where* Setup_λ *outputs a public parameter* $\mathsf{pp} \in PP_\lambda$.
- $\mathcal{S}amp\mathcal{Y}es = \{\mathsf{SampYes}_\lambda : PP_\lambda \to R_\lambda\}$ *where* $\mathsf{SampYes}_\lambda$ *on input* $\mathsf{pp} \in PP_\lambda$ *outputs a random element* $x \in L_\lambda$ *with a witness* $w \in W_\lambda$*, i.e., a random element* $(x, w) \in R_\lambda$.
- $\mathcal{S}amp\mathcal{N}o = \{\mathsf{SampNo}_\lambda : PP_\lambda \to X_\lambda \setminus L_\lambda\}$ *where* SampNo_λ *on input* $\mathsf{pp} \in PP_\lambda$ *outputs a random element* $x \in X_\lambda \setminus L_\lambda$.
- $\mathcal{K}ey\mathcal{G}en = \{\mathsf{KeyGen}_\lambda : PP_\lambda \to PK_\lambda \times SK_\lambda\}$ *where* KeyGen_λ *on input* $\mathsf{pp} \in PP_\lambda$ *outputs a public key* pk *and secret key* sk *such that* $pk = \alpha_\lambda(sk)$.
- $\mathcal{P}riv = \{\mathsf{Priv}_\lambda : PP_\lambda \times SK_\lambda \times X_\lambda \to \mathit{\Pi}_\lambda\}$ *where* Priv_λ *on input* $\mathsf{pp} \in PP_\lambda$, $sk \in SK_\lambda$*, and an instance* $x \in X_\lambda$ *outputs its proof* $\pi = H_\lambda(sk, x)$.
- $\mathcal{P}ub = \{\mathsf{Pub}_\lambda : PP_\lambda \times PK_\lambda \times R_\lambda \to \mathit{\Pi}_\lambda\}$ *where* Pub_λ *on input* $\mathsf{pp} \in PP_\lambda$, $pk \in PK_\lambda$*, and an instance with a witness* $(x, w) \in R_\lambda$ *outputs its proof* $\pi \in \mathit{\Pi}_\lambda$.

$(\mathcal{S}etup, \mathcal{S}amp\mathcal{Y}es, \mathcal{S}amp\mathcal{N}o, \mathcal{K}ey\mathcal{G}en, \mathcal{P}riv, \mathcal{P}ub)$ *is a* $\mathcal{C}_1$-hash proof system (HPS) against $\mathcal{C}_2$ *if for any* $\lambda \in \mathbb{N}$*, it holds that:*

- **Correctness***: For any* $(x, w) \in R_\lambda$*, we have*

$$\mathsf{Priv}_\lambda(\mathsf{pp}, sk, x) = H_\lambda(sk, x) = \mathsf{Pub}_\lambda(\mathsf{pp}, pk, x, w)$$

  where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$ *and* $(pk, sk) \leftarrow \mathsf{KeyGen}_\lambda(\mathsf{pp})$.
- **Subset membership problem***:*
  - *The distributions of* x *and* x' *are identical where* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(x, w) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$*, and* $x' \leftarrow L_\lambda$.
  - *The distributions of* x *and* x' *are identical where* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $x \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$*, and* $x' \leftarrow X_\lambda \setminus L_\lambda$.
  - *For any* $\mathcal{G} = \{g_\lambda\} \in \mathcal{C}_2$,

$$|\Pr[g_\lambda(\mathsf{pp}, x_0) = 1] - \Pr[g_\lambda(\mathsf{pp}, x_1) = 1]| \leq \mathsf{negl}(\lambda)$$

    where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(x_0, w) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$*, and* $x_1 \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$.

$(\mathcal{S}etup, \mathcal{S}amp\mathcal{Y}es, \mathcal{S}amp\mathcal{N}o, \mathcal{K}ey\mathcal{G}en, \mathcal{P}riv, \mathcal{P}ub)$ *is* perfectly smooth $\mathcal{C}_1$-HPS against $\mathcal{C}_2$ *if it satisfies the following property.*

- **Perfect smoothness***: For any* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$*, the following random variables are identical, i.e.,* 0*-close.*

$$(x, pk, \pi),\ (x, pk, \pi')$$

  where $\mathbf{x} \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$, $(pk, sk) \leftarrow \mathsf{KeyGen}_\lambda(\mathsf{pp})$, $\pi = \mathsf{Priv}_\lambda(\mathsf{pp}, sk, x)$*, and* $\pi' \leftarrow \mathit{\Pi}$.

$(\mathcal{S}etup, \mathcal{S}amp\mathcal{Y}es, \mathcal{S}amp\mathcal{N}o, \mathcal{K}ey\mathcal{G}en, \mathcal{P}riv, \mathcal{P}ub)$ *is* ϵ-universal$_1$ $\mathcal{C}_1$-HPS against $\mathcal{C}_2$ *if it satisfies the following property.*

- **ϵ-universality$_1$**: *For any* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $pk \in PK_\lambda$, $x \in X_\lambda \backslash L_\lambda$ *and* $\pi \in \Pi_\lambda$, *it holds that*
$$\Pr[\mathsf{Priv}_\lambda(\mathsf{pp}, sk, x) = \pi \mid \alpha_\lambda(sk) = pk] \leq \epsilon.$$
If ϵ *is a negligible function, then* $(\mathcal{S}etup,\ \mathcal{S}amp\mathcal{Y}es,\ \mathcal{S}amp\mathcal{N}o,\ \mathcal{K}ey\mathcal{G}en,\ \mathcal{P}riv,\ \mathcal{P}ub)$ *is a* strong universal$_1$ $\mathcal{C}_1$-HPS against $\mathcal{C}_2$.

$(\mathcal{S}etup, \mathcal{S}amp\mathcal{Y}es, \mathcal{S}amp\mathcal{N}o, \mathcal{K}ey\mathcal{G}en, \mathcal{P}riv, \mathcal{P}ub)$ *is* ϵ-universal$_2$ $\mathcal{C}_1$-HPS against $\mathcal{C}_2$ *if it satisfies the following property.*

- **ϵ-universality$_2$**: *For any* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $pk \in PK_\lambda$, $x, x^* \in X_\lambda$ *and* $\pi, \pi^* \in \Pi_\lambda$ *with* $x \notin L_\lambda \cup \{x^*\}$, *it holds that*
$$\Pr[\mathsf{Priv}_\lambda(\mathsf{pp}, sk, x) = \pi \mid \mathsf{Priv}_\lambda(\mathsf{pp}, sk, x^*) = \pi^* \wedge \alpha_\lambda(sk) = pk] \leq \epsilon.$$
If ϵ *is a negligible function, then* $(\mathcal{S}etup,\ \mathcal{S}amp\mathcal{Y}es,\ \mathcal{S}amp\mathcal{N}o,\ \mathcal{K}ey\mathcal{G}en,\ \mathcal{P}riv,\ \mathcal{P}ub)$ *is a* strong universal$_2$ $\mathcal{C}_1$-HPS against $\mathcal{C}_2$.

Definition 8 (Trapdoor One-Way Function) *Let* $\mathcal{K}ey\mathcal{G}en = \{\mathsf{KeyGen}_\lambda : \phi \to EK_\lambda \times TK_\lambda\}$, $\mathcal{E}val = \{\mathsf{Eval}_\lambda : EK_\lambda \times \mathcal{D}_\lambda \to \mathcal{R}_\lambda\}$ *and* $\mathcal{I}nverse = \{\mathsf{Inverse}_\lambda : TK_\lambda \times \mathcal{D}_\lambda \to \mathcal{R}_\lambda\}$ *be function families where* $\mathcal{D}_\lambda$ *and* $\mathcal{R}_\lambda$ *are determined by the key pair* (ek, tk) *generated by* KeyGen_λ. $(\mathcal{K}ey\mathcal{G}en, \mathcal{E}val, \mathcal{I}nverse)$ *is a* $\mathcal{C}_1$-trapdoor one-way function (TDF) against $\mathcal{C}_2$ *if:*

- **Correctness:** *For any* $\lambda \in \mathbb{N}$, *any* $(ek, tk) \leftarrow \mathsf{KeyGen}_\lambda$, *and any* $X \in \mathcal{D}_\lambda$*:*
$$\mathsf{Inverse}_\lambda(tk, \mathsf{Eval}_\lambda(ek, X)) = X.$$
- **One-wayness:** *For any* $\mathcal{G} = \{g_\lambda\} \in \mathcal{C}_2$, *and any* $\lambda \in \mathbb{N}$*:*
$$\Pr\left[\mathsf{Eval}_\lambda(ek, g_\lambda(ek, Y)) = Y \;\middle|\; \begin{array}{c} (ek, tk) \leftarrow \mathsf{KeyGen}_\lambda \\ X \leftarrow \mathcal{D}_\lambda \\ Y = \mathsf{Eval}_\lambda(ek, X) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

2.4 Sampling Procedure

In this section, we recall the sampling procedure in [17], and then show several lemmas on the sampling procedure that will be used later in the security proofs.

Construction 1 (Sampling Procedure) *Let* $\mathbf{M}_0^n$ *and* $\mathbf{M}_1^n$ *be the following* $n \times n$ *matrices:*
$$\mathbf{M}_0^n = \begin{pmatrix} 0 & & \cdots & 0 & 0 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 0 & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix},\ \mathbf{M}_1^n = \begin{pmatrix} 0 & & \cdots & 0 & 1 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 0 & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}.$$

- $\mathsf{LSamp}(n)$*:*
  1. *Output the following* $n \times n$ *upper triangular matrix:*

$$\begin{pmatrix} 1 & r_{1,2} & \cdots & r_{1,n-1} & r_{1,n} \\ 0 & 1 & r_{2,3} & \cdots & r_{2,n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1,n} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

  where $r_{i,j} \leftarrow \{0,1\}$*.*
- $\mathsf{RSamp}(n)$*:*
  1. *Output the following* $n \times n$ *matrix:*

$$\begin{pmatrix} 1 & & \cdots & 0 & r_1 \\ 0 & 1 & & & r_2 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

  where $r_i \leftarrow \{0,1\}$*.*
- $\mathsf{ZeroSamp}$*(n):*
  1. *Sample* $\mathbf{R}_1 \leftarrow \mathsf{LSamp}(n)$ *and* $\mathbf{R}_2 \leftarrow \mathsf{RSamp}(n)$*.*
  2. *Output* $\mathbf{R}_1\mathbf{M}_0^n\mathbf{R}_2$*.*
- $\mathsf{OneSamp}$*(n):*
  1. *Sample* $\mathbf{R}_1 \leftarrow \mathsf{LSamp}(n)$ *and* $\mathbf{R}_2 \leftarrow \mathsf{RSamp}(n)$*.*
  2. *Output* $\mathbf{R}_1\mathbf{M}_1^n\mathbf{R}_2$*.*

Here, the output of $\mathsf{ZeroSamp}(n)$ *is always a matrix of rank* $n-1$ *and the output of* $\mathsf{OneSamp}(n)$ *is always a matrix of full rank.*

Lemma 2 ([31, 5]) *If* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$*, then there is a polynomial* n *such that for any family* $\mathcal{F} = \{f_\lambda\}$ *in* NC^1 *and any* $\lambda \in \mathbb{N}$*, we have*

$$\begin{aligned} &|\Pr[f_\lambda(\mathbf{M}) = 1 \mid \mathbf{M} \leftarrow \mathsf{ZeroSamp}(n(\lambda))] - \\ &\qquad \Pr[f_\lambda(\mathbf{M}') = 1 \mid \mathbf{M}' \leftarrow \mathsf{OneSamp}(n(\lambda))]| \leq \mathsf{negl}(\lambda). \end{aligned}$$

Lemma 3 *For any* $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(n)$*, it holds that* $\mathrm{Ker}(\mathbf{M}) = \{\mathbf{0}, \mathbf{k}\}$ *where* $\mathbf{k}$ *is a vector such that* $\mathbf{k} \in \{0,1\}^{n-1} \times 1$*.*

Proof. $\mathbf{M}$ is a matrix sampled from $\mathsf{ZeroSamp}(n)$, i.e.,

$$\begin{aligned} \mathbf{M} &= \mathbf{R}_1\mathbf{M}_0^n\mathbf{R}_2 \\ &= \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 0 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 0 & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & & \cdots & 0 & r_1 \\ 0 & 1 & & & r_2 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

where $\mathbf{R}_1 \leftarrow \mathsf{LSamp}(n)$ and $\mathbf{R}_2 \leftarrow \mathsf{RSamp}(n)$. Then, we have $\mathbf{k} = (r_1 r_2 \cdots 1)^\top \in \mathrm{Ker}(\mathbf{M})$ since

$$\begin{aligned}
\mathbf{M} &= \mathbf{R}_1 \mathbf{M}_0^n \mathbf{R}_2 \mathbf{k} \\
&= \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 0 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 0 & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & & \cdots & 0 & r_1 \\ 0 & 1 & & & r_2 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & r_{n-1} \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n-1} \\ 1 \end{pmatrix} \\
&= \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 0 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 0 & \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = \mathbf{R}_1 \mathbf{0} = \mathbf{0}.
\end{aligned}$$

Moreover, according to Lemma 1, there are only two vectors $\boldsymbol{v}$ such that $\mathbf{M}\boldsymbol{v} = \mathbf{0}$. Therefore, we have $\mathrm{Ker}(\mathbf{M}) = \{\mathbf{0}, \mathbf{k}\}$, completing the proof of Lemma 3. □

Lemma 4 *For any* $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(n)$*, it holds that* $\mathrm{Ker}(\mathbf{M}^\top) = \{\mathbf{0}, \mathbf{k}\}$ *where* $\mathbf{k}$ *is a vector such that* $\mathbf{k} \in 1 \times \{0,1\}^{n-1}$.

Proof. $\mathbf{M}$ is a matrix sampled from $\mathsf{ZeroSamp}(n)$ i.e., $\mathbf{M} = \mathbf{R}_1 \mathbf{M}_0^n \mathbf{R}_2$, where $\mathbf{R}_1 \leftarrow \mathsf{LSamp}(n)$, $\mathbf{R}_2 \leftarrow \mathsf{RSamp}(n)$. Since $\mathbf{R}_1^\top$ has full rank, the equation $\mathbf{R}_1^\top \mathbf{x} = (1\ 0\ \cdots\ 0)^\top$ has a unique solution $\mathbf{x}^*$. $\mathbf{x}^*$ is in the kernel of $\mathbf{M}^\top$ since $\mathbf{R}_2^\top \mathbf{M}_0^{n\top} \mathbf{R}_1^\top \mathbf{x}^* = \mathbf{R}_2^\top \mathbf{M}_0^{n\top} (1\ 0\ \cdots\ 0)^\top = \mathbf{R}_2^\top \mathbf{0} = \mathbf{0}$. According to the following equation

$$\mathbf{R}_1^\top \mathbf{x}^* = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ r_{2,1} & 1 & 0 & \cdots & 0 \\ r_{3,1} & r_{3,2} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ r_{n,1} & \cdots & & r_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} x_1^* \\ x_2^* \\ x_3^* \\ \vdots \\ x_n^* \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

we have $x_1^* = 1$, i.e., $\mathbf{x} \in 1 \times \{0,1\}^{n-1}$.

Moreover, according to Lemma 1 and the fact that the rank of $\mathbf{M}^\top$ is $n-1$, there are only two vectors $\boldsymbol{v}$ such that $\mathbf{M}^\top \boldsymbol{v} = \mathbf{0}$. Therefore, we have $\mathrm{Ker}(\mathbf{M}^\top) = \{\mathbf{0}, \mathbf{x}^*\}$, completing the proof of Lemma 4. □

Lemma 5 *For any* $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$*, it holds that*

$$\mathrm{Im}(\mathbf{M}^\top) = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\} = \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}.$$

Proof. Let $U = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$ and $V = \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$. Let $\mathbf{k}$ be a non-zero vector such that $\mathbf{k} \in \mathrm{Ker}(\mathbf{M}^\top)$.

According to Lemma 4, we have $\mathbf{k} \in 1 \times \{0,1\}^{\lambda-1}$. Therefore, for any $\mathbf{x} \in U$ such that $\mathbf{x} = \mathbf{M}^\top \mathbf{w}$ where $\mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}$, we have $\mathbf{x} = \mathbf{M}^\top \mathbf{w} = \mathbf{M}^\top(\mathbf{w}+\mathbf{k}) \in V$ since $(\mathbf{w}+\mathbf{k}) \in 1 \times \{0,1\}^{\lambda-1}$. Moreover, for any $\mathbf{x} \in V$ such that $\mathbf{x} = \mathbf{M}^\top \mathbf{w}$ where $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}$, we have $\mathbf{x} = \mathbf{M}^\top \mathbf{w} = \mathbf{M}^\top(\mathbf{w}+\mathbf{k}) \in U$ since $(\mathbf{w}+\mathbf{k}) \in 0 \times \{0,1\}^{\lambda-1}$. Therefore, we have $U = V$ and it follows that $\mathrm{Im}(\mathbf{M}^\top) = U \cup V = U \cup U = U = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$. In the same way, we have $\mathrm{Im}(\mathbf{M}^\top) = U \cup V = V \cup V = V = \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$. As a result, we have

$$\mathrm{Im}(\mathbf{M}^\top) = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\} = \{\mathbf{x}' | \mathbf{w}' \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\},$$

completing the proof of Lemma 5. □

Lemma 6 *For any* $\mathbf{M} \in \mathsf{ZeroSamp}(\lambda)$*, the distributions of* $\mathbf{x}$ *and* $\mathbf{x}'$ *are identical where* $\mathbf{w} \leftarrow 0 \times \{0,1\}^{\lambda-1}$, $\mathbf{w}' \leftarrow 0 \times \{0,1\}^{\lambda-1}$, $\mathbf{x} = \mathbf{M}^\top \mathbf{w}$*, and* $\mathbf{x}' = \mathbf{M}^\top \mathbf{w}'$.

Proof. Let $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and let $U = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$ and $V = \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$. According to Lemma 5, we have $U = V = \mathrm{Im}(\mathbf{M}^\top)$. Moreover, since the rank of $\mathbf{M}$ is $\lambda - 1$, we have $|U| = |V| = |\mathrm{Im}(\mathbf{M}^\top)| = 2^{\lambda-1}$. It follows that for any $\mathbf{y} \in \mathrm{Im}(\mathbf{M}^\top)$, there exists and only exist $\mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}$ (resp. $\mathbf{w}' \in 1 \times \{0,1\}^{\lambda-1}$) such that $\mathbf{y} = \mathbf{M}^\top \mathbf{w}$ (resp. $\mathbf{y} = \mathbf{M}^\top \mathbf{w}'$). Then, Lemma 6 immediately follows.

Lemma 7 *The distributions of* $\mathbf{M} + \mathbf{N}$ *and* $\mathbf{M}'$ *are identical, where* $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$*,* $\mathbf{M}' \leftarrow \mathsf{OneSamp}(\lambda)$*, and* $\mathbf{N}$ *is the following matrix.*

$$\mathbf{N} = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Proof. For $\mathbf{R}_1 \leftarrow \mathsf{LSamp}(\lambda)$ and $\mathbf{R}_2 \leftarrow \mathsf{RSamp}(\lambda)$, we have

$$\begin{aligned}
\mathbf{M}' = \mathbf{R}_1 \mathbf{M}_1^\lambda \mathbf{R}_2 &= \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 1 \\ 1 & 0 & & \vdots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \mathbf{R}_2 \\
&= \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 0 \\ 1 & 0 & & \vdots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \mathbf{R}_2 + \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 1 \\ 0 & 0 & & \vdots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \mathbf{R}_2 \\
&= \mathbf{R}_1 \begin{pmatrix} 0 & & \cdots & 0 & 0 \\ 1 & 0 & & \vdots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} \mathbf{R}_2 + \begin{pmatrix} 0 & & \cdots & 0 & 1 \\ 0 & 0 & & \vdots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \\
&= \mathbf{R}_1 \mathbf{M}_0^\lambda \mathbf{R}_2 + \mathbf{N} = \mathbf{M} + \mathbf{N}.
\end{aligned}$$

Hence, the distributions of $\mathbf{M}+\mathbf{N}$ and $\mathbf{M}'$ are identical for $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M}' \leftarrow \mathsf{OneSamp}(\lambda)$, completing the proof of Lemma 7. □

3 Construction of NC^1-OWP against NC^1

In this section, we first give our construction of a collection of NC^1-OWPs against NC^1 under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. Next, we extend it to a NC^1-OWP against NC^1 based on the same assumption.

Construction 2 (Collection of NC^1-OWPs) *Let λ be a security parameter. We define the families $\mathcal{K}ey\mathcal{G}en = \{\mathsf{KeyGen}_\lambda\}$ with key spaces $\{K_\lambda = \{\mathbf{M} \mid \mathbf{M} \in \mathsf{OneSamp}(\lambda)\}\}$ and $\mathcal{E}val = \{\mathsf{Eval}_\lambda\}$ as follows.*

- KeyGen_λ*:*
  1. *Sample $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$.*
  2. *Output $\mathbf{M}$ (which defines the domain as $D_{\lambda,\mathbf{M}} := \{0,1\}^\lambda$).*
- $\mathsf{Eval}_\lambda(\mathbf{M},\mathbf{x})$*:*
  1. *Compute $\mathbf{y} := \mathbf{M}\mathbf{x}$ and output $\mathbf{y}$.*

Theorem 1 *$(\mathcal{K}ey\mathcal{G}en, \mathcal{E}val)$ defined as Construction 2 is a collection of NC^1-OWPs against NC^1 under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$.*

Proof sketch. As described in Introduction, our construction is essentially a "lossy function". More specifically, it is straightforward that our scheme is a permutation, since $\mathbf{M}$ is of full rank when $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$. Moreover, when we generate $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ instead of $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ in KeyGen_λ, we can prove that an adversary $\mathcal{A}$ breaking the one-wayness of our construction with probability ϵ can also be used to find a second pre-image $\mathbf{x}'$ for $\mathsf{Eval}_\lambda(\mathbf{M},\mathbf{x})$ such that $\mathbf{x} \neq \mathbf{x}'$ with probability $\frac{1}{2}\epsilon$. This is due to the fact that $\mathbf{M}$ is not of full rank in this case and $\mathcal{A}$ has no information on whether the pre-image is $\mathbf{x}$ or $\mathbf{x}'$. However, it is unlikely that $\mathcal{A}$ can find such a second pre-image, since this construction is indistinguishable with the original one, where $\mathbf{M}$ is generated as $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ and there exists no second pre-image for each $\mathbf{M}$. Therefore, we can conclude that this scheme is one-way, which immediately gives us the one-wayness of the original scheme (due to the indistinguishability between $\mathsf{OneSamp}(\lambda)$ and $\mathsf{ZeroSamp}(\lambda)$).

The formal proof is as follows.

Proof. First note that both $\mathcal{K}ey\mathcal{G}en$ and $\mathcal{E}val$ are computable in NC^1, since they only involve operations including multiplications of a constant number of matrices, inner products, and sampling random bits. We now show that $(\mathcal{K}ey\mathcal{G}en, \mathcal{E}val)$ satisfies computability and one-wayness.

Permutation. Since for $\mathbf{M} \leftarrow \mathsf{OneSamp}_\lambda$, $\mathbf{M}$ is a full rank matrix, we have that $\mathsf{Eval}_\lambda(\mathbf{M},\mathbf{x}) = \mathbf{M}\mathbf{x} \in D_{\lambda,\mathbf{M}} = \{0,1\}^\lambda$ is a permutation.

One-wayness. Let $\mathcal{A} = \{a_\lambda\}$ be any adversary in NC^1. We give hybrid games to show that the advantage of $\mathcal{A}$ in breaking the one-wayness of Construction 2 is negligible.

Game 0: This is the original one-wayness game for $\mathcal{A} = \{a_\lambda\}$. $\mathcal{CH}$ runs $\mathbf{M} \leftarrow \mathsf{KeyGen}_\lambda$ and samples $\mathbf{x} \leftarrow \{0,1\}^\lambda$. Then, it runs $\mathbf{y} = \mathsf{Eval}_\lambda(\mathbf{M}, \mathbf{x})$ and sends $\mathbf{y}$ to a_λ. a_λ succeeds if it outputs $\tilde{\mathbf{x}}$ such that $\mathbf{x} = \tilde{\mathbf{x}}$. Otherwise, it fails.

Game 1: This game is the same as **Game 0** except that $\mathcal{CH}$ runs $\mathsf{ZeroSamp}(\lambda)$ instead of $\mathsf{OneSamp}(\lambda)$ in the key generation procedure.

Lemma 8 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_0 (resp., ϵ_1) in* **Game 0** *(resp.,* **Game 1***), then $|\epsilon_0 - \epsilon_1| = \mathsf{negl}(\lambda)$.*

Proof. We now construct $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$ that distinguishes $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ with advantage $|\epsilon_0 - \epsilon_1|$, which contradicts to Lemma 2.

b_λ takes as input $\mathbf{M}$, which is generated as $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ from its challenger. Then, it samples $\mathbf{x} \leftarrow \{0,1\}^\lambda$. Next, b_λ runs $\mathbf{y} = \mathsf{Eval}_\lambda(\mathbf{M}, \mathbf{x})$ and sends $\mathbf{y}$ to a_λ. When a_λ outputs $\tilde{\mathbf{x}}$, if $\mathbf{x} = \tilde{\mathbf{x}}$, b_λ outputs 1. Otherwise, it outputs 0.

Since all operations in b_λ are performed in NC^1, we have $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$.

One can see that when $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ (resp., $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$), the view of a_λ is identical to its view in **Game 0** (resp., **Game 1**), i.e., b_λ outputs 1 with probability ϵ_0 (resp., ϵ_1). Therefore, $\mathcal{B} = \{b_\lambda\}$ distinguishes $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ with advantage $|\epsilon_0 - \epsilon_1|$, which should be negligible according to Lemma 2, completing the proof of Lemma 8. □

Game 2: This game is the same as **Game 1** except that a_λ succeeds if $\mathbf{x} \neq \tilde{\mathbf{x}} \wedge \mathsf{Eval}_\lambda(\mathbf{M}, \mathbf{x}) = \mathsf{Eval}(\mathbf{M}, \tilde{\mathbf{x}})$.

Lemma 9 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_1 (resp., ϵ_2) in* **Game 1** *(resp.,* **Game 2***), then $\epsilon_1 = \epsilon_2$.*

Proof. According to Lemma 1 and due to the fact that the rank of $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ is $\lambda - 1$, for any $\mathbf{y} \in \mathrm{Im}(\mathbf{M})$, there are two vectors $\mathbf{x}, \mathbf{x}'$ such that $\mathbf{Mx} = \mathbf{Mx}' = \mathbf{y} \wedge \mathbf{x} \neq \mathbf{x}'$, and we have

$$\begin{aligned}
\epsilon_1 &= \Pr\left[\tilde{\mathbf{x}} = \mathbf{x}^* \;\middle|\; \begin{array}{c} \mathbf{x}^* \leftarrow \{\mathbf{x}, \mathbf{x}'\} \\ \mathbf{y} = \mathbf{Mx}^* \\ \tilde{\mathbf{x}} \leftarrow a_\lambda(\mathbf{y}) \end{array}\right] \\
&= \frac{1}{2}\Pr\left[\tilde{\mathbf{x}} = \mathbf{x} \;\middle|\; \begin{array}{c} \mathbf{x}^* = \mathbf{x} \\ \mathbf{y} = \mathbf{Mx}^* \\ \tilde{\mathbf{x}} \leftarrow a_\lambda(\mathbf{y}) \end{array}\right] + \frac{1}{2}\Pr\left[\tilde{\mathbf{x}} = \mathbf{x}' \;\middle|\; \begin{array}{c} \mathbf{x}^* = \mathbf{x}' \\ \mathbf{y} = \mathbf{Mx}^* \\ \tilde{\mathbf{x}} \leftarrow a_\lambda(\mathbf{y}) \end{array}\right] \\
&= \frac{1}{2}\Pr\left[\tilde{\mathbf{x}} = \mathbf{x} \;\middle|\; \begin{array}{c} \mathbf{x}^* = \mathbf{x}' \\ \mathbf{y} = \mathbf{Mx}^* \\ \tilde{\mathbf{x}} \leftarrow a_\lambda(\mathbf{y}) \end{array}\right] + \frac{1}{2}\Pr\left[\tilde{\mathbf{x}} = \mathbf{x}' \;\middle|\; \begin{array}{c} \mathbf{x}^* = \mathbf{x} \\ \mathbf{y} = \mathbf{Mx}^* \\ \tilde{\mathbf{x}} \leftarrow a_\lambda(\mathbf{y}) \end{array}\right] \\
&= \Pr\left[\begin{array}{c} \tilde{\mathbf{x}} \neq \mathbf{x}^* \wedge \\ \mathsf{Eval}_\lambda(\mathbf{M}, \tilde{\mathbf{x}}) = \mathsf{Eval}_\lambda(\mathbf{M}, \mathbf{x}^*) \end{array} \;\middle|\; \begin{array}{c} \mathbf{x}^* \leftarrow \{\mathbf{x}, \mathbf{x}'\} \\ \mathbf{y} = \mathbf{Mx}^* \\ \tilde{\mathbf{x}} \leftarrow a_\lambda(\mathbf{y}) \end{array}\right] = \epsilon_2,
\end{aligned}$$

completing the proof of Lemma 9.

Lemma 10 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_2 in* **Game 2**, *then $\epsilon_2 = \mathsf{negl}(\lambda)$.*

Proof. We now construct $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$ that distinguishes $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ with advantage ϵ_2, which contradicts to Lemma 2.

b_λ takes as input $\mathbf{M}$, which is generated as $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ from its challenger. Then, it samples $\mathbf{x} \leftarrow \{0,1\}^\lambda$. Next, b_λ runs $\mathbf{y} = \mathsf{Eval}_\lambda(\mathbf{M}, \mathbf{x})$ and send $\mathbf{y}$ to a_λ. When a_λ outputs $\tilde{\mathbf{x}}$, if $\mathbf{x} \neq \tilde{\mathbf{x}} \wedge \mathbf{y} = \mathsf{Eval}_\lambda(\mathbf{M}, \mathbf{x})$, b_λ outputs 1. Otherwise, it outputs 0.

Since all operations in b_λ are performed in NC^1, we have $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$.

One can see that when $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$, the view of a_λ is identical to its view in **Game 2**, i.e., b_λ outputs 1 with probability ϵ_2.

When $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$, since $\mathsf{Eval}(\mathbf{M}, \tilde{\mathbf{x}})$ is permutation, there is no vector $\tilde{\mathbf{x}}$ such that $\mathbf{x} \neq \tilde{\mathbf{x}} \wedge \mathbf{y} = \mathsf{Eval}_\lambda(\mathbf{k}, \mathbf{x})$, i.e. b_λ outputs 1 with probability 0.

Therefore, $\mathcal{B} = \{b_\lambda\}$ distinguishes $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ with advantage ϵ_2, which should be negligible according to Lemma 2, completing the proof of Lemma 10. □

Since $|\epsilon_0 - \epsilon_1| = \mathsf{negl}(\lambda)$, $\epsilon_1 = \epsilon_2$, and $\epsilon_2 = \mathsf{negl}(\lambda)$, we have

$$\epsilon_0 \leq |\epsilon_0 - \epsilon_1| + \epsilon_1 = \mathsf{negl}(\lambda) + \epsilon_2 = \mathsf{negl}(\lambda),$$

i.e., Construction 2 satisfies one-wayness. This completes the proof of Theorem 1. □

Extension to NC^1-OWPs against NC^1. We now show a transformation from collections of NC^1-OWPs against NC^1, where the output distributions of the key generation algorithms are uniformly random over key space, to NC^1-OWPs against NC^1. Specifically, given a collection of OWPs $\{f_k : D_k \to D_k\}_{k \in K}$ where K is the key space, we construct a OWP $g : D \to D$ where $D := \bigcup_{k \in K}(\{k\} \times$

D_k) and $g((k,x) \in D) = (k, f_k(x))$. This transformation can be applied in NC^1, and the properties of permutation and one-wayness of g hold due to those properties of f. Note that in [5], it is shown that $\mathsf{OneSamp}(\lambda)$ samples $\mathbf{M} \leftarrow \{\mathbf{M} \in \mathsf{OneSamp}(\lambda)\}$ uniformly. Thus, KeyGen_λ of our construction samples $k \leftarrow K_\lambda = \{\mathbf{M} \mid \mathbf{M} \in \mathsf{OneSamp}(\lambda)\}$ uniformly, and we can apply this transformation to our collection of NC^1-OWPs against NC^1. We refer the reader to Section A in the supplementary material for the details.

Computability in $\mathsf{AC}^0[2]$. Perhaps interestingly, our one-way permutation can be run by an even smaller class of circuits $\mathsf{AC}^0[2]$, which satisfies $\mathsf{AC}^0[2] \subsetneq \mathsf{NC}^1$ [41, 43] and consists of constant-depth circuits with $\mathsf{MOD2}$ gates. The reason is that it only involves multiplications of a constant number of matrices, inner products, and sampling random bits. Due to the same reason, our constructions of single-bit HPS introduced later in Section 4 is also computable in $\mathsf{AC}^0[2]$.

4 Construction of NC^1-HPS against NC^1

In this section, we start by giving a construction of perfectly smooth and $\frac{1}{2}$-universal$_1$ NC^1-HPS against NC^1 such that the proof space is one-bit. Next, we turn this construction into a perfectly smooth and strong universal$_1$ NC^1-HPS against NC^1 such that the proof space is multi-bit. Finally, we construct a strong universal$_2$ NC^1-HPS against NC^1 such that the language L supports $\{0,1\}^n$.

4.1 Perfectly Smooth and Universal$_1$ for One-Bit

In this section, we give our construction of perfectly smooth and $\frac{1}{2}$-universal$_1$ NC^1-HPS against NC^1 circuits under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$.

Construction 3 **(NC^1-HPS)** *Let λ be a security parameter. We define the families $\mathcal{S}etup = \{\mathsf{Setup}_\lambda\}$, $\mathcal{S}amp\mathcal{Y}es = \{\mathsf{SampYes}_\lambda\}$, $\mathcal{S}amp\mathcal{N}o = \{\mathsf{SampNo}_\lambda\}$, $\mathcal{K}ey\mathcal{G}en = \{\mathsf{KeyGen}_\lambda\}$, $\mathcal{P}riv = \{\mathsf{Priv}_\lambda\}$, and $\mathcal{P}ub = \{\mathsf{Pub}_\lambda\}$ as follows.*

- Setup_λ:
  1. *Sample* $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$.
  2. *Output* $\mathsf{pp} = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, \Pi_\lambda, H_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$ *where*
     - $X_\lambda := \{0,1\}^\lambda$.
     - $L_\lambda := \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\} = \mathrm{Im}(\mathbf{M}^\top)$ *($\because$ Lemma 5).*
     - $W_\lambda := 1 \times \{0,1\}^{\lambda-1}$.
     - $R_\lambda := \{(\mathbf{x}, \mathbf{w}) |\ \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}\mathbf{w}\}$.
     - $SK_\lambda := \{0,1\}^\lambda$.
     - $PK_\lambda := \mathrm{Im}(\mathbf{M})$.
     - $\Pi_\lambda := \{0,1\}$.
     - $H_\lambda(\mathbf{sk}, \mathbf{x}) := \mathbf{sk}^\top \mathbf{x}$.
     - $\alpha_\lambda(\mathbf{sk}) := \mathbf{M}\,\mathbf{sk}$.
     - $\mathsf{aux}_\lambda := \mathbf{M}$.
- $\mathsf{SampYes}_\lambda(\mathsf{pp})$:

1. *Parse* $\mathsf{pp} = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, \Pi_\lambda, H_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$ *and let* $\mathsf{aux}_\lambda = \mathbf{M}$.
2. *Sample* $\mathbf{w} \leftarrow 1 \times \{0,1\}^{\lambda-1}$.
3. *Compute* $\mathbf{x} := \mathbf{M}^\top \mathbf{w}$ *and output* $\mathbf{x}$.

- $\mathsf{SampNo}_\lambda(\mathsf{pp})$*:*
  1. *Parse* $\mathsf{pp} = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, \Pi_\lambda, H_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$ *and let* $\mathsf{aux}_\lambda = \mathbf{M}$.
  2. *Sample* $\mathbf{w} \leftarrow 1 \times \{0,1\}^{\lambda-1}$.
  3. *Compute* $\mathbf{M}'$ *as*
  $$\mathbf{M}' = \mathbf{M} + \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}.$$
  4. *Compute* $\mathbf{x} := \mathbf{M}'^\top \mathbf{w}$ *and output* $\mathbf{x}$.
- $\mathsf{KeyGen}_\lambda(\mathsf{pp})$*:*
  1. *Parse* $\mathsf{pp} = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, H_\lambda, \Pi_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$.
  2. *Sample* $\mathbf{sk} \leftarrow SK_\lambda$.
  3. *Compute* $\mathbf{pk} := \alpha_\lambda(\mathbf{sk})$ *and output* $(\mathbf{pk}, \mathbf{sk})$.
- $\mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x})$*:*
  1. *Parse* $\mathsf{pp} = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, \Pi_\lambda, H_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$.
  2. *Compute* $\pi := H_\lambda(\mathbf{sk}, \mathbf{x})$ *and output* π.
- $\mathsf{Pub}_\lambda(\mathsf{pp}, \mathbf{pk}, \mathbf{x}, \mathbf{w})$*:*
  1. *Compute* $\pi := \mathbf{pk}^\top \mathbf{w}$ *and output* π.

Theorem 2 *If* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$*, then (*$\mathcal{S}etup$*,* $\mathcal{S}amp\mathcal{Y}es$*,* $\mathcal{S}amp\mathcal{N}o$*,* $\mathcal{K}ey\mathcal{G}en$*,* $\mathcal{P}riv$*,* $\mathcal{P}ub$*) defined as Construction 3 is a perfectly smooth and* $\frac{1}{2}$*-universal*$_1$ NC^1*-HPS against* NC^1 *circuits.*

Proof sketch. It is straightforward that this HPS is correct.

To show the subset membership problem of our construction, we first give two observations: (1) for any $\mathbf{M}$ sampled from $\mathsf{ZeroSamp}(\lambda)$, the distribution of $\mathbf{M} + \mathbf{N}$ is identical to $\mathsf{OneSamp}(\lambda)$, where

$$\mathbf{N} = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix},$$

and (2) perhaps interestingly, for any $\mathbf{w} \in 0 \times \{0,1\}^{n-1}$ (respectively, $\mathbf{w} \in 1 \times \{0,1\}^{n-1}$), there is a vector $\mathbf{k}$ in the kernel of $\mathbf{M}$ such that $\mathbf{M}\mathbf{w}^\top = \mathbf{M}(\mathbf{w}+\mathbf{k})^\top$ and $(\mathbf{w}+\mathbf{k}) \in 1 \times \{0,1\}^{n-1}$ (respectively, $(\mathbf{w}+\mathbf{k}) \in 0 \times \{0,1\}^{n-1}$), which implies $\mathrm{Im}(\mathbf{M}^\top) = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{n-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\} = \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^{n-1}, \mathbf{x} = \mathbf{M}^\top \mathbf{w}\}$. Since for any vector $\mathbf{w} \in 0 \times \{0,1\}^{n-1}$, it holds that $(\mathbf{M}+\mathbf{N})^\top \mathbf{w} = \mathbf{M}^\top \mathbf{w} + \mathbf{N}^\top \mathbf{w} = \mathbf{M}^\top \mathbf{w} + \mathbf{0} = \mathbf{M}^\top \mathbf{w}$, we have

$$L = \mathrm{Im}(\mathbf{M}^\top) = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\}$$

due to observation (2). We now switch the distribution of $\mathbf{M}$ from $\mathsf{ZeroSamp}(\lambda)$ to $\mathsf{OneSamp}(\lambda)$ by exploiting the indistinguishability between $\mathsf{ZeroSamp}(\lambda)$ and $\mathsf{OneSamp}(\lambda)$. Then one can see that the language is now

$$\begin{aligned} L' &= \{\mathbf{x}|\mathbf{w} \in 0 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\} \\ &= \{\mathbf{x}|\mathbf{w} \in 1 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\}, \end{aligned}$$

since $(\mathbf{M}+\mathbf{N})^\top \in \mathsf{ZeroSamp}(\lambda)$ now. Next we return the distribution of $\mathbf{M}$ to $\mathsf{ZeroSamp}(\lambda)$ by exploiting the indistinguishability between $\mathsf{ZeroSamp}(\lambda)$ and $\mathsf{OneSamp}(\lambda)$ again, and the language becomes exactly $X \setminus L$. The reason is that when $(\mathbf{M}+\mathbf{N})^\top \in \mathsf{ZeroSamp}(\lambda)$, we have

$$X = \{0,1\}^n = \{\mathbf{x}|\mathbf{w} \in \{0,1\}^n, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\}$$

since $\mathbf{M}+\mathbf{N}$ is of full rank due to observation (1), i.e., we have

$$\begin{aligned} X \setminus L &= \{\mathbf{x}|\mathbf{w} \in \{0,1\}^n, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\} \setminus \{\mathbf{x}|\mathbf{w} \in 0 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\} \\ &= \{\mathbf{x}|\mathbf{w} \in 1 \times \{0,1\}^{n-1}, \mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}\}. \end{aligned}$$

As a result, L and $X \setminus L$ are indistinguishable in NC^1.

We now explain the intuition of the proof of universal$_1$. Since the rank of $\mathbf{M}$ is $n-1$, when we fix the public key $\mathbf{pk}$, there are two different secret keys $\mathbf{sk}$ and $\mathbf{sk}'$ such that $\mathbf{pk} = \mathbf{M}\,\mathbf{sk} = \mathbf{M}\,\mathbf{sk}'$. As explained before, for any $\mathbf{x} \in X \setminus L$, there exists $\mathbf{w} \in 1 \times \{0,1\}^{n-1}$ such that $\mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}$, and $(\mathbf{M}+\mathbf{N})$ is a full rank matrix. Therefore, we have $(\mathbf{M}+\mathbf{N})\mathbf{sk} \neq (\mathbf{M}+\mathbf{N})\mathbf{sk}'$ which implies $\mathbf{N}\,\mathbf{sk} \neq \mathbf{N}\,\mathbf{sk}'$, i.e., either $\mathbf{N}\,\mathbf{sk}$ or $\mathbf{N}\,\mathbf{sk}'$ is zero-vector and the other is $(1\ 0 \cdots 0)^\top$. Therefore, when we let $\mathbf{N}\,\mathbf{sk} = (0 \cdots 0)^\top$ and $\mathbf{N}\,\mathbf{sk}' = (1\ 0 \cdots 0)^\top$, it holds that $H(\mathbf{sk}, \mathbf{x}) = \mathbf{sk}^\top (\mathbf{M}+\mathbf{N})^\top \mathbf{w} = \mathbf{sk}^\top \mathbf{M}^\top \mathbf{w} + (0 \cdots 0)\mathbf{w} = \mathbf{sk}^\top \mathbf{M}^\top \mathbf{w}$ and $H(\mathbf{sk}', \mathbf{x}) = \mathbf{sk}'^\top (\mathbf{M}+\mathbf{N})^\top \mathbf{w} = \mathbf{sk}'^\top \mathbf{M}^\top \mathbf{w} + (1\ 0 \cdots 0)\mathbf{w} = \mathbf{sk}'^\top \mathbf{M}^\top \mathbf{w} + 1$, which implies $H(\mathbf{sk}, \mathbf{x}) \neq H(\mathbf{sk}', \mathbf{x})$. As a result, for fixed $\mathbf{pk}$, one can guess the proof for an instance $\mathbf{x} \in X \setminus L$ with probability at most $\frac{1}{2}$ since there is no information on whether the secret key is $\mathbf{sk}$ or $\mathbf{sk}'$.

The formal proof is as follows.

Proof. First note that all of the algorithms $\mathcal{S}etup$, $\mathcal{S}amp\mathcal{Y}es$, $\mathcal{S}amp\mathcal{N}o$, $\mathcal{K}ey\mathcal{G}en$, $\mathcal{P}riv$, and $\mathcal{P}ub$ are in NC^1, since they only involve operations including multiplications of a constant number of matrices, inner products, and sampling random bits.

Next we prove that Construction 3 satisfies correctness, subset membership problem, perfect smoothness, and $\frac{1}{2}$-universality$_1$.

Correctness. Since $\mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x}) = H_\lambda(\mathbf{sk}, \mathbf{x}) = \mathbf{sk}^\top \mathbf{x} = \mathbf{sk}^\top \mathbf{M}^\top \mathbf{w} = (\mathbf{M}\mathbf{sk})^\top \mathbf{w} = \mathbf{pk}^\top \mathbf{w} = \mathsf{Pub}_\lambda(\mathsf{pp}, \mathbf{pk}, \mathbf{x}, \mathbf{w})$, Construction 3 satisfies correctness.

Subset membership problem. We now propose and prove three propositions corresponding to the three properties in the definition of subset membership problem (see Definition 7) respectively.

Proposition 1 *The distributions of* $\mathbf{x}$ *and* $\mathbf{x}'$ *are identical where* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$*,* $\mathbf{x} \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$*, and* $\mathbf{x}' \leftarrow L_\lambda$*.*

Proof. Let $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ be a matrix generated in the procedure of Setup_λ. Let f be a map $f : 1 \times \{0,1\}^{\lambda-1} \to \mathrm{Im}(\mathbf{M}^\top)$ such that $f(\mathbf{w}) = \mathbf{M}^\top\mathbf{w}$. One can see that for any $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, the distributions of $\mathbf{x}$ and $\mathbf{x}'$ are identical where $\mathbf{x} \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$, $\mathbf{w}' \leftarrow 1 \times \{0,1\}^{\lambda-1}$, and $\mathbf{x}' = f(\mathbf{w}')$. Moreover, if f is bijective, the distributions of $\mathbf{x}'$ and $\mathbf{x}''$ are identical for $\mathbf{w}' \leftarrow 1 \times \{0,1\}^{\lambda-1}$, $\mathbf{x}' = f(\mathbf{w}')$, and $\mathbf{x}'' \leftarrow \mathrm{Im}(\mathbf{M}^\top)$. Therefore, if f is bijective, the distributions of $\mathbf{x}$ and $\mathbf{x}''$ are identical. Namely, to show Proposition 1, we only have to show that f is bijective.

Injectivity. We now show that for any $\mathbf{w}, \mathbf{w}' \leftarrow 1 \times \{0,1\}^{\lambda-1}$ such that $\mathbf{w} \neq \mathbf{w}'$, we have $f(\mathbf{w}) \neq f(\mathbf{w}')$. We prove by contradiction, i.e. we show that if there are $\mathbf{w}, \mathbf{w}' \leftarrow 1 \times \{0,1\}^{\lambda-1}$ such that $\mathbf{w} \neq \mathbf{w}'$ and $f(\mathbf{w}) = f(\mathbf{w}')$, then it contradicts on Lemma 4.

Since $\mathbf{M}^\top\mathbf{w} = \mathbf{M}^\top\mathbf{w}'$, we have $\mathbf{M}^\top(\mathbf{w} - \mathbf{w}') = \mathbf{0}$. Moreover, since $\mathbf{w} \neq \mathbf{w}'$ and $\mathbf{w}, \mathbf{w}' \in 1 \times \{0,1\}^{\lambda-1}$, $\mathbf{w} - \mathbf{w}'$ is the non-zero vector in the kernel of $\mathbf{M}^\top$ and $\mathbf{w} - \mathbf{w}' \in 0 \times \{0,1\}^{\lambda-1}$. However, according to Lemma 4, we have $\mathrm{Ker}(\mathbf{M}^\top) = \{\mathbf{0}, \mathbf{k}\}$ where $\mathbf{k} \in 1 \times \{0,1\}^\lambda$, which gives us the conflict.

Surjectivity. We now show that for any $\mathbf{x} \in \mathrm{Im}(\mathbf{M}^\top)$, there exists a vector $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}$ such that $\mathbf{x} = f(\mathbf{w})$, i.e., $\mathbf{x} = \mathbf{M}^\top\mathbf{w}$. According to Lemma 5, we have $\mathrm{Im}(\mathbf{M}^\top) = \{\mathbf{x} | \mathbf{w} \in 1 \times \{0,1\}^\lambda, \mathbf{x} = \mathbf{M}^\top\mathbf{w}\}$. Therefore, it holds that for any $\mathbf{x} \in \mathrm{Im}(\mathbf{M}^\top)$, there exists $\mathbf{w} \in 1 \times \{0,1\}^\lambda$ such that $\mathbf{x} = \mathbf{M}^\top\mathbf{w}$, i.e., $\mathbf{x} = f(\mathbf{w})$, completing the proof of surjectivity.

Putting all the above together, Proposition 1 immediately follows. □

Proposition 2 *The distributions of* $\mathbf{x}$ *and* $\mathbf{x}'$ *are identical for* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$*,* $\mathbf{x} \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$*, and* $\mathbf{x}' \leftarrow X_\lambda \setminus L_\lambda$*.*

Proof. Let $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M}'$ be

$$\mathbf{M}' = \mathbf{M} + \begin{pmatrix} 0 \cdots & 0 & 1 \\ 0 \;\; 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 \cdots & 0 & 0 \end{pmatrix}.$$

We first show that for any $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}$, we have $\mathbf{M}'^\top\mathbf{w} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$.

($\bigstar$) For any $\mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}$, we have

$$\mathbf{M}'\mathbf{w} = \left(\mathbf{M} + \begin{pmatrix} 0 \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 \cdots & 0 & 0 \end{pmatrix} \right)^{\top} \mathbf{w}$$

$$= \mathbf{M}^{\top}\mathbf{w} + \begin{pmatrix} 0 \cdots & & 0 \\ \vdots & 0 & \cdots & 0 \\ 0 & & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 0 \\ w_2 \\ \vdots \\ w_\lambda \end{pmatrix} = \mathbf{M}^{\top}\mathbf{w} + \mathbf{0} = \mathbf{M}^{\top}\mathbf{w}.$$

Moreover, according to Lemma 5, we have $\mathrm{Im}(\mathbf{M}^{\top}) = \{\mathbf{x} | \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^{\top}\mathbf{w}\}$. Hence, we have

$$\begin{aligned}\{\mathbf{x} \mid \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}'^{\top}\mathbf{w}\} &= \{\mathbf{x} \mid \mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}^{\top}\mathbf{w}\} \;(\bigstar) \\ &= \mathrm{Im}(\mathbf{M}^{\top}).\end{aligned}$$

As a result, for all $\mathbf{x} \in \mathrm{Im}(\mathbf{M}^{\top})$, there exists $\mathbf{w}' \in 0 \times \{0,1\}^{\lambda-1}$ such that $\mathbf{x} = \mathbf{M}'^{\top}\mathbf{w}'$. Moreover, according to Lemma 7, $\mathbf{M}'$ is a full rank matrix, which means that for any $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}$ and any $\mathbf{x} \in \mathrm{Im}(\mathbf{M}^{\top})$, we have $\mathbf{M}'^{\top}\mathbf{w} \neq \mathbf{x}$. Namely, for any $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}$, we have $\mathbf{M}'^{\top}\mathbf{w} \in \{0,1\}^{\lambda} \setminus \mathrm{Im}(\mathbf{M}^{\top})$.

It is straightforward that for any $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, the distributions of $\mathbf{x} \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$ and $\mathbf{x}' = \mathbf{M}'^{\top}\mathbf{w}'$ are identical where $\mathbf{w}' \leftarrow 1 \times \{0,1\}^{\lambda-1}$. Moreover, since $\mathbf{M}'^{\top}$ is of full rank, the map $f : 1 \times \{0,1\}^{n-1} \to \{0,1\}^{\lambda} \setminus \mathrm{Im}(\mathbf{M}^{\top})$ such that $f(\mathbf{w}) = \mathbf{M}'^{\top}\mathbf{w}$ is bijective, i.e., the distributions of $\mathbf{x}' = f(\mathbf{w}')$ and $\mathbf{x}'' \leftarrow \{0,1\}^{\lambda} \setminus \mathrm{Im}(\mathbf{M}^{\top})$ are identical for $\mathbf{w}' \leftarrow 1 \times \{0,1\}^{\lambda-1}$, completing the proof of Proposition 2. □

Proposition 3 *For any $\mathcal{A} = \{a_\lambda\} \in \mathsf{NC}^1$,*

$$|\Pr[a_\lambda(\mathsf{pp}, \mathbf{x}_0) = 1] - \Pr[a_\lambda(\mathsf{pp}, \mathbf{x}_1) = 1]| \leq \mathsf{negl}(\lambda)$$

where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(\mathbf{x}_0, \mathbf{w}) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$, *and* $\mathbf{x}_1 \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$.

Proof. Let $\mathcal{A} = \{a_\lambda\}$ be any adversary in NC^1 and $\mathbf{N}$ be the following $\lambda \times \lambda$ matrix.

$$\mathbf{N} = \begin{pmatrix} 0 \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 \cdots & 0 & 0 \end{pmatrix}.$$

We give hybrid games to show that the advantage of $\mathcal{A}$ in breaking the hardness of subset membership problem is negligible.

Game 0: This is the original $\mathsf{SampYes}$ game for $\mathcal{A}$. $\mathcal{CH}$ runs $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(\mathbf{x}, \mathbf{w}) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$. Then it sends $(\mathsf{pp}, \mathbf{x})$ to a_λ. a_λ succeeds if a_λ outputs 1. Otherwise, it fails.

Game 1: This game is the same as **Game 0** except that $\mathcal{CH}$ samples $\mathbf{w} \leftarrow 0 \times \{0,1\}^{\lambda-1}$ in the procedure of $\mathsf{SampYes}_\lambda(\mathsf{pp})$.

Lemma 11 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_0 (resp., ϵ_1) in* **Game 0** *(resp.,* **Game 1***), then $\epsilon_0 = \epsilon_1$.*

Proof. Lemma 11 follows from the fact that the distributions of $\mathbf{x}$ and $\mathbf{x}'$ are identical where $\mathbf{M} \leftarrow \mathsf{ZeroSamp}_\lambda$, $\mathbf{w} \leftarrow 1 \times \{0,1\}^{\lambda-1}$, $\mathbf{x} = \mathbf{M}^\top \mathbf{w}$, $\mathbf{w}' \leftarrow 0 \times \{0,1\}^{\lambda-1}$, and $\mathbf{x}' = \mathbf{M}^\top \mathbf{w}'$ (according to Lemma 6). □

Game 2: This game is the same as **Game 1** except that $\mathcal{CH}$ computes $\mathbf{x} := (\mathbf{M} + \mathbf{N})^\top \mathbf{w}$ in the procedure of $\mathsf{SampYes}_\lambda(\mathsf{pp})$.

Lemma 12 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_1 (resp., ϵ_2) in* **Game 1** *(resp.,* **Game 2***), then $\epsilon_1 = \epsilon_2$.*

Proof. Lemma 12 follows from the fact that for $\mathbf{w} \in 0 \times \{0,1\}^{\lambda-1}$, we have

$$\mathbf{x} = (\mathbf{M} + \mathbf{N})^\top \mathbf{w} = \mathbf{M}^\top \mathbf{w} + \mathbf{N}^\top \mathbf{w} = \mathbf{M}^\top \mathbf{w}.$$

□

Game 3: This game is the same as **Game 2** except that $\mathcal{CH}$ runs $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ in the procedure of Setup_λ.

Lemma 13 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_2 (resp., ϵ_3) in* **Game 2** *(resp.,* **Game 3***), then $|\epsilon_2 - \epsilon_3| = \mathsf{negl}(\lambda)$.*

Proof. We now construct $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$ that distinguishes $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ with advantage $|\epsilon_2 - \epsilon_3|$, which contradicts to Lemma 2.

b_λ takes as input $\mathbf{M}$, which is generated as $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ or $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ from its challenger. Then, it runs $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$ using $\mathbf{M}$, samples $\mathbf{w} \leftarrow 0 \times \{0,1\}^{\lambda-1}$, and sets $x := (\mathbf{M} + \mathbf{N})^\top \mathbf{w}$. Next, b_λ gives $(\mathsf{pp}, \mathbf{x})$ to a_λ. When a_λ outputs b, then b_λ outputs b.

Since all operations in b_λ are performed in NC^1, we have $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$.

One can see that when $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ (resp., $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$), the view of a_λ is identical to its view in **Game 2** (resp., **Game 3**), i.e., b_λ outputs 1 with probability ϵ_2 (resp., ϵ_3). Therefore, $\mathcal{B} = \{b_\lambda\}$ distinguishes $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$ with advantage $|\epsilon_2 - \epsilon_3|$, which should be negligible according to Lemma 2, completing the proof of Lemma 13. □

Game 4: This game is the same as **Game 3** except that $\mathcal{CH}$ samples $\mathbf{w} \leftarrow 1 \times \{0,1\}^{\lambda-1}$ in the procedure of $\mathsf{SampYes}_\lambda(\mathsf{pp})$.

Lemma 14 *If* $\mathcal{A} = \{a_\lambda\}$ *succeeds with advantage* ϵ_3 *(resp.,* ϵ_4*) in* **Game 3** *(resp.,* **Game 4***), then* $\epsilon_3 = \epsilon_4$*.*

Proof. Let $\mathbf{M} \leftarrow \mathsf{OneSamp}(\lambda)$. According to Lemma 7, we have $\mathbf{M} + \mathbf{N} \in \mathsf{ZeroSamp}(\lambda)$. Therefore, according to Lemma 6, the distributions of $\mathbf{x}$ and $\mathbf{x}'$ are identical where $\mathbf{w} \leftarrow 1 \times \{0,1\}^{\lambda-1}$, $\mathbf{x} = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}$, $\mathbf{w}' \leftarrow 0 \times \{0,1\}^{\lambda-1}$, and $\mathbf{x}' = (\mathbf{M}+\mathbf{N})^\top \mathbf{w}'$, completing the proof of Lemma 14.

□

Game 5: This game is the same as **Game 4** except that $\mathcal{CH}$ runs $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ in the procedure of Setup_λ, which is identical to the original SampNo game.

Lemma 15 *If* $\mathcal{A} = \{a_\lambda\}$ *succeeds with advantage* ϵ_4 *(resp.,* ϵ_5*) in* **Game 4** *(resp.,* **Game 5***), then* $|\epsilon_4 - \epsilon_5| = \mathsf{negl}(\lambda)$*.*

Proof. We can prove Lemma 15 in the same way as Lemma 11.

Note that, $\epsilon_0 = \Pr[a_\lambda(\mathsf{pp}, \mathbf{x}) = 1]$ and $\epsilon_5 = \Pr[a_\lambda(\mathsf{pp}, \mathbf{x}') = 1]$ where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(\mathbf{x}, \mathbf{w}) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$, and $\mathbf{x}' \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$. Moreover, since $\epsilon_0 = \epsilon_1 = \epsilon_2$, $|\epsilon_2 - \epsilon_3| = \mathsf{negl}(\lambda)$, $\epsilon_3 = \epsilon_4$, and $|\epsilon_4 - \epsilon_5| = \mathsf{negl}(\lambda)$ we have

$$|\epsilon_0 - \epsilon_5| = |\epsilon_2 - \epsilon_5| \leq |\epsilon_2 - \epsilon_3| + |\epsilon_3 - \epsilon_4| + |\epsilon_4 - \epsilon_5| = \mathsf{negl}(\lambda).$$

□

According to Proposition 1, 2, and 3, Construction 3 satisfies the subset membership problem, completing this part of proof.

Perfect smoothness. We now show that for any $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, the random variables $(\mathbf{x}, \mathbf{pk}, \pi)$ and $(\mathbf{x}, \mathbf{pk}, \pi')$ are identical where $\mathbf{x} \leftarrow X_\lambda \setminus L_\lambda$, $(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathsf{KeyGen}_\lambda(\mathsf{pp})$, and $\pi' \leftarrow \Pi_\lambda$.

According to Lemma 1, for any $\mathbf{pk}^* \in PK_\lambda$, there are only two secret keys $\mathbf{sk}$ and $\mathbf{sk}'$ such that $\mathbf{pk}^* = \alpha_\lambda(\mathbf{sk}) = \mathbf{M}\,\mathbf{sk} = \alpha_\lambda(\mathbf{sk}') = \mathbf{M}\,\mathbf{sk}'$. Moreover, according to Lemma 3, we have $\mathbf{sk} = \mathbf{sk}' + \mathbf{k}$ where $\mathbf{k}$ is a vector such that $\mathbf{k} \in \mathrm{Ker}(\mathbf{M})$ and $\mathbf{k} \in \{0,1\}^{\lambda-1} \times 1$, i.e., the last elements in $\mathbf{sk}$ and $\mathbf{sk}'$ are different (one is 1 and other is 0). Therefore, for any $\mathbf{x}^* \in X_\lambda \setminus L_\lambda$ and $\mathbf{pk}^* \in PK_\lambda$, we have

$$\begin{aligned}
\pi = \mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x}^*) &= \mathbf{sk}^\top \mathbf{M}'^\top \mathbf{w}^* \\
&= \mathbf{sk}^\top \left(\mathbf{M} + \begin{pmatrix} 0 \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 \cdots & 0 & 0 \end{pmatrix} \right)^\top \mathbf{w}^* \\
&= \mathbf{sk}^\top \mathbf{M}^\top \mathbf{w}^* + (sk_1 \ \cdots \ sk_\lambda) \begin{pmatrix} 0 \cdots & & 0 \\ \vdots & 0 & \cdots & 0 \\ 0 & & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 \\ w_2^* \\ \vdots \\ w_\lambda^* \end{pmatrix} \\
&= {\mathbf{pk}^*}^\top \mathbf{w}^* + sk_\lambda,
\end{aligned}$$

and it follows that for any $\mathbf{x}^* \in X_\lambda \setminus L_\lambda$ and $\mathbf{pk}^* \in PK_\lambda$, there are two secret key $\mathbf{sk}, \mathbf{sk}'$ such that $\mathbf{pk}^* = \alpha_\lambda(\mathbf{sk}) = \alpha_\lambda(\mathbf{sk}')$, $\mathsf{Priv}_\lambda(\mathbf{sk}, \mathbf{x}^*) = 0$, and $\mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}', \mathbf{x}^*) = 1$. Namely, the number of secret keys satisfying $\mathbf{pk}^* = \alpha_\lambda(\mathbf{sk}) \ \wedge \ \pi^* = \mathsf{Priv}_\lambda(\mathbf{sk}, \mathbf{x}^*)$ is 1. Therefore, we have

$$\begin{aligned}
\Pr[(\mathbf{x}, \mathbf{pk}, \pi) = (\mathbf{x}^*, \mathbf{pk}^*, \pi^*)] &= \Pr\left[\begin{matrix} \mathbf{pk} = \mathbf{pk}^* \wedge \\ \pi = \pi^* \end{matrix} \middle| \ \mathbf{x} = \mathbf{x}^* \right] \Pr[\mathbf{x} = \mathbf{x}^*] \\
&= \Pr\left[\begin{matrix} \mathbf{pk}^* = \alpha_\lambda(\mathbf{sk}) \wedge \\ \pi^* = \mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x}^*) \end{matrix} \right] \Pr\left[\mathbf{x} = \mathbf{x}^*\right] \\
&= \frac{1}{|SK_\lambda|} \Pr\left[\mathbf{x} = \mathbf{x}^*\right]
\end{aligned}$$

where $\mathbf{sk} \leftarrow SK_\lambda$ and $\mathbf{x} \leftarrow X_\lambda \times L_\lambda$. Similarly, we have

$$\begin{aligned}
\Pr[(\mathbf{x}, \mathbf{pk}, \pi') = (\mathbf{x}^*, \mathbf{pk}^*, \pi^*)] &= \Pr[\pi' = \pi^*] \Pr[\mathbf{pk} = \mathbf{pk}^*] \Pr[\mathbf{x} = \mathbf{x}^*] \\
&= \frac{1}{2} \frac{2}{|SK_\lambda|} \Pr\left[\mathbf{x} = \mathbf{x}^*\right] \\
&= \frac{1}{|SK_\lambda|} \Pr\left[\mathbf{x} = \mathbf{x}^*\right].
\end{aligned}$$

Therefore, we have $\Pr[(\mathbf{x}, \mathbf{pk}, \pi) = (\mathbf{x}^*, \mathbf{pk}^*, \pi^*)] = \Pr[(\mathbf{x}, \mathbf{pk}, \pi') = (\mathbf{x}^*, \mathbf{pk}^*, \pi^*)]$ and it follows that Construction 3 satisfies perfect smoothness.

$\frac{1}{2}$-universality$_1$. $\frac{1}{2}$-universality$_1$ follows from the fact that for any $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $\mathbf{x} \in X_\lambda \setminus L_\lambda$, $\mathbf{pk} \in PK_\lambda$, and $\pi \in \Pi_\lambda$, the number of secret keys such that $\mathbf{pk} = \alpha_\lambda(\mathbf{sk})$ is 2 and the number of secret keys such that $\mathbf{pk} = \alpha_\lambda(\mathbf{sk}) \ \wedge \ \pi =$

$\mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x})$ is 1 as described above. Therefore, we have

$$\begin{aligned}\Pr[\mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x}) = \pi \wedge \alpha_\lambda(\mathbf{sk}) = \mathbf{pk}] &= \frac{1}{|SK_\lambda|} = \frac{1}{2}\frac{2}{|SK_\lambda|} \\ &= \frac{1}{2}\Pr[\alpha_\lambda(\mathbf{sk}) = \mathbf{pk}] \\ \Leftrightarrow \Pr[\mathsf{Priv}_\lambda(\mathsf{pp}, \mathbf{sk}, \mathbf{x}) = \pi | \alpha_\lambda(\mathbf{sk}) = \mathbf{pk}] &= \frac{1}{2}.\end{aligned}$$

Therefore, Construction 3 satisfies $\frac{1}{2}$-universality$_1$.

Putting all the above together, Theorem 2 immediately follows. □

Multi-bit NC^1-HPS. Notice that the size of proof space of Construction 3 is only one-bit, which makes it less useful. However, we can extend this construction with multi-bit proofs by running multiple HPS in parallel. We refer the reader to Section B in the supplementary material for the multi-bit version of our HPS and the security proof.

Universal$_2$ NC^1-HPS. By carefully adopting the technique by Cramer and Shoup [16], we achieve a universal$_2$ NC^1-HPS. The resulting scheme can be computed in NC^1 and it is secure against NC^1 circuits under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$. We refer the reader to Section C in the supplementary material for the details.

4.2 Application: NC^1-CCA Secure PKE

As one of the most important application of HPSs, Cramer and shoup [16] constructed a CCA secure PKE scheme. Interestingly, by instantiating the underlying HPS with our construction, we immediately achieve an NC^1-CCA secure PKE scheme against NC^1 circuits restricted in the same way as the ones defined for verifiable computation schemes by Campanelli and Gennaro [14], i.e., ones allowed to make constant rounds of adaptive queries to the decryption oracle, while in each round, they can make arbitrary polynomial number of queries. We refer the reader to Section D in the supplementary material for the details on this application.

5 Construction of NC^1-TDF against NC^1

In this section, we give our construction of NC^1-TDF against NC^1 under the assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$.

Construction 4 (NC^1-TDF) *Let λ be a security parameter and l be a polynomial in λ. Let $\mathcal{F} = \{f_\lambda : \{0,1\}^\lambda \to \{0,1\}^{l(\lambda)}\}$ be a NC^1-OWF against NC^1. We define the families $\mathcal{KeyGen} = \{\mathsf{KeyGen}_\lambda\}$ with key spaces $EK_\lambda = \{\mathbf{M} \mid \mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)\}$ and $TK_\lambda = \mathrm{Ker}(\mathbf{M})$, $\mathcal{Eval} = \{\mathsf{Eval}_\lambda\}$ and $\mathcal{Inverse} = \{\mathsf{Inverse}_\lambda\}$ as follows.*

- KeyGen_λ*:*
  1. *Run* $\mathbf{R} \leftarrow \mathsf{LSamp}(\lambda)$*,* $\mathbf{R}' \leftarrow \mathsf{RSamp}(\lambda)$*.*
  2. *Set* $\mathbf{k} := (\boldsymbol{r}\ 1)^\top$ *where* $(\boldsymbol{r}\ 1)^\top$ *is the last column of* $\mathbf{R}'$*.*
  3. *Compute* $\mathbf{M} := \mathbf{R}\mathbf{M}_0^\lambda\mathbf{R}'$ *where* $\mathbf{M}_0^\lambda$ *is defined as Construction 1.*
  4. *Set* $ek := \mathbf{M}$ *and* $tk := \mathbf{k}$*, and output* (ek, tk) *(according to the proof of Lemma 3, it holds that* $\mathbf{k} \in \mathrm{Ker}(\mathbf{M})$ *).*

  The domain $\mathcal{D}_{\lambda,ek}$ *is defined as follows.*

$$\mathcal{D}_{\lambda,ek} := \{0,1\}^\lambda \times \left(\mathrm{Im}(\mathbf{M}^\top) \times \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)\right)^\lambda.$$

  The range $\mathcal{R}_{\lambda,ek}$ *is defined as follows.*

$$\mathcal{R}_{\lambda,ek} := \{Y \mid \exists X \in \mathcal{D}_{\lambda,ek} \ s.t.\ Y = \mathsf{Eval}_\lambda(ek, X)\}.$$

- $\mathsf{Eval}_\lambda(ek, X)$*:*
  1. *Parse* $X := (x, (\mathbf{c}_{1,0}, \mathbf{c}_{1,1}), (\mathbf{c}_{2,0}, \mathbf{c}_{2,1}), \cdots, (\mathbf{c}_{\lambda,0}, \mathbf{c}_{\lambda,1})) \in \mathcal{D}_{\lambda,ek}$*.*
  2. *For* $x = x_1x_2\cdots x_\lambda \in \{0,1\}^\lambda$ *and all* $i \in [\lambda]$*, if* $x_i = 0$*, set* $(\mathbf{c}_i, \mathbf{c}'_i) := (\mathbf{c}_{i,0}, \mathbf{c}_{i,1})$*, otherwise set* $(\mathbf{c}_i, \mathbf{c}'_i) := (\mathbf{c}_{i,1}, \mathbf{c}_{i,0})$*.*
  3. *Compute* $y := f_\lambda(x)$*.*
  4. *Set* $Y := (y, (\mathbf{c}_1, \mathbf{c}'_1), (\mathbf{c}_2, \mathbf{c}'_2), \cdots, (\mathbf{c}_\lambda, \mathbf{c}'_\lambda))$ *and output* Y*.*
- $\mathsf{Inverse}_\lambda(tk, Y)$*:*
  1. *Parse* $tk := \mathbf{k}$ *and* $Y := (y, (\mathbf{c}_1, \mathbf{c}'_1), (\mathbf{c}_2, \mathbf{c}'_2), \cdots, (\mathbf{c}_\lambda, \mathbf{c}'_\lambda)) \in \mathcal{R}_{\lambda,ek}$*.*
  2. *For all* $i \in [\lambda]$*, if* $\mathbf{k}^\top\mathbf{c}_i = 0 \wedge \mathbf{k}^\top\mathbf{c}'_i = 1$*, then set* $x_i := 0$ *and* $(\mathbf{c}_{i,0}, \mathbf{c}_{i,1}) := (\mathbf{c}_i, \mathbf{c}'_i)$*.*
  3. *Else if* $\mathbf{k}^\top\mathbf{c}_i = 1 \wedge \mathbf{k}^\top\mathbf{c}'_i = 0$*, then set* $x_i := 1$ *and* $(\mathbf{c}_{i,0}, \mathbf{c}_{i,1}) := (\mathbf{c}'_i, \mathbf{c}_i)$*.*
  4. *Else output* $\perp$ *and halt.*
  5. *Set* $X = (x, (\mathbf{c}_{1,0}, \mathbf{c}_{1,1}), \cdots, (\mathbf{c}_{\lambda,0}, \mathbf{c}_{\lambda,1}))$ *and output* X*.*

Theorem 3 *If* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L/poly}$ *and there exists an* NC^1*-OWF against* NC^1 *circuits, Construction 4 is an* NC^1*-TDF against* NC^1*.*

Proof sketch. Let $\mathbf{k}$ and $\mathbf{M}$ be the trapdoor key and evaluation key generated by KeyGen_λ respectively. For any $\mathbf{c} \in \mathrm{Im}(\mathbf{M}^\top)$, we must have $\mathbf{k}^\top\mathbf{c} = 0$ since $\mathbf{k} \in \mathrm{Ker}(\mathbf{M})$. Also, we prove that for any $\mathbf{c} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$, there must exists $\mathbf{w}$ such that $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}$ and $\mathbf{c} = \left(\mathbf{M} + \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}\right)\mathbf{w}$. Since $\mathbf{k}^\top\mathbf{M}^\top\mathbf{w} = 0$ and $\mathbf{k}^\top \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}^\top \mathbf{w} = 1$, we must have $\mathbf{k}^\top\mathbf{c} = 1$ in this case. Therefore, $\mathbf{k}$, which is samplable in NC^1, can be used to determine whether $\mathbf{c}$ is in $\mathrm{Im}(\mathbf{M}^\top)$ or $\{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ and recover x_i by checking whether $\mathbf{c}_i$ and $\mathbf{c}'_i$ are swapped in the inversing procedure, i.e., correctness holds.

Moreover, due to the subset membership problem for $L = \mathrm{Im}(\mathbf{M})$, the uniform distributions over $\mathrm{Im}(\mathbf{M}^\top)$ and $\{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ are indistinguishable when $\mathbf{M}$ is a correctly generated evaluation key, i.e., the distributions $(\mathbf{c}_{i,0}, \mathbf{c}_{i,1}) \leftarrow \mathrm{Im}(\mathbf{M}^\top) \times \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ and $(\mathbf{c}_{i,0}, \mathbf{c}_{i,1}) \leftarrow \mathrm{Im}(\mathbf{M}^\top) \times \mathrm{Im}(\mathbf{M}^\top)$ are indistinguishable. Therefore, the adversary in the one-way game can only obtain information on $f_\lambda(x)$ (which is one-way), and the additional pairs $(\mathbf{c}_{i,0}, \mathbf{c}_{i,1})$ can be simulated by just sampling them from $\mathrm{Im}(\mathbf{M}^\top) \times \mathrm{Im}(\mathbf{M}^\top)$, i.e., they reveal little information on x.

The formal proof is as follows.

Proof. Note that $\mathcal{KeyGen}$, $\mathcal{E}val$, and $\mathcal{I}nverse$ only involve operations including multiplications of the constant number of matrices, inner products and sampling random bits. Since these operation can be performed in NC^1, we have $\mathcal{KeyGen}$, $\mathcal{E}val$, and $\mathcal{I}nverse$ can be computed in NC^1.

Next, we prove that Construction 4 satisfies correctness and one-wayness.

Correctness. For any $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and any $\boldsymbol{c} \in \mathrm{Im}(\mathbf{M}^\top)$, we have

$$\mathbf{k}^\top \boldsymbol{c} = \mathbf{k}^\top \mathbf{M}^\top \mathbf{w} = (\mathbf{M}\mathbf{k})^\top \mathbf{w} = \mathbf{0}^\top \mathbf{w} = 0$$

where $\mathbf{k} \in \mathrm{Ker}(\mathbf{M})$ and $\mathbf{w}$ is a vector such that $\boldsymbol{c} = \mathbf{M}\mathbf{w}$.

Next we show that when $\boldsymbol{c} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ then $\mathbf{k}^\top \mathbf{c} = 1$. Before showing this, we first propose the following lemma, which is straightforwardly implied by Proposition 2 in Theorem 2.

Lemma 16 *For any* $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$*, it holds that*

$$\{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top) = \{\mathbf{x} \mid \exists \mathbf{w} \in 1 \times \{0,1\}^{\lambda-1}, \mathbf{x} = \mathbf{M}'^\top \mathbf{w}\}$$

where

$$\mathbf{M}' = \mathbf{M} + \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}.$$

According to Lemma 16, for any $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and any $\boldsymbol{c} \in \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$, we have

$$\begin{aligned}
\mathbf{k}^\top \boldsymbol{c} = \mathbf{k}^\top \mathbf{M}'^\top \mathbf{w} = \mathbf{k}^\top \left(\mathbf{M} + \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix} \right)^\top \mathbf{w} \\
= (\mathbf{M}\mathbf{k})^\top \mathbf{w} + \left(\begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix} \mathbf{k} \right)^\top \mathbf{w} \\
= \mathbf{0}^\top \mathbf{w} + (1\ 0\ \cdots\ 0)\mathbf{w} = 0 + 1 = 1
\end{aligned}$$

where $\mathbf{k}$ is a vector in the kernel of $\mathbf{M}$ and $\mathbf{w}$ is a vector such that $\mathbf{w} \in 1 \times \{0,1\}^{\lambda-1} \wedge \boldsymbol{c} = \mathbf{M}'^{\top}\mathbf{w}$.

As a result, for all $i \in [\lambda]$, $\mathbf{k}$ generated by KeyGen_λ can be used to determine whether $\boldsymbol{c}_i$ (resp., $\boldsymbol{c}_i'$) generated by Eval_λ are in $\mathrm{Im}(\mathbf{M}^\top)$ or $\{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$, and hence recover x_i.

One-wayness. Let $\mathcal{A} = \{a_\lambda\}$ be any adversary in NC^1. We give hybrid games to show that the advantage of $\mathcal{A}$ in breaking the one-wayness of Construction 4 is negligible.

Game 0: This is the original one-wayness game for $\mathcal{A} = \{a_\lambda\}$. $\mathcal{CH}$ runs $(ek, tk) \leftarrow \mathsf{KeyGen}_\lambda$, samples $X \leftarrow \mathcal{D}_{\lambda,ek}$, and runs $Y = \mathsf{Eval}_\lambda(ek, X)$. Then it sends (ek, Y) to a_λ. a_λ succeeds if a_λ outputs X^* such that $\mathsf{Eval}_\lambda(ek, X^*) = Y$. Otherwise it fails.

Game 1 $\sim$ Game $\boldsymbol{\lambda}$: For $i \in [\lambda]$, **Game i** is the same as **Game i-1** except that $\mathcal{CH}$ samples $(\mathbf{c}_{i,0}, \mathbf{c}_{i,1}) \leftarrow \mathrm{Im}(\mathbf{M}^\top) \times \mathrm{Im}(\mathbf{M}^\top)$.

Lemma 17 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_{i-1} (resp., ϵ_i) in* **Game i-1** *(resp.,* **Game i***), then $|\epsilon_{i-1} - \epsilon_i| = \mathsf{negl}(\lambda)$.*

Proof. According to the proof of the part of subset membership problem in Theorem 2, we have the following lemma.

Lemma 18 *For any $\mathcal{G} = \{g_\lambda\} \in \mathsf{NC}^1$,*

$$|\Pr[g_\lambda(\mathbf{M}, \mathbf{c}_0) = 1] - \Pr[g_\lambda(\mathbf{M}, \mathbf{c}_1) = 1]| \leq \mathsf{negl}(\lambda),$$

where $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$, $\mathbf{c}_0 \leftarrow \mathrm{Im}(\mathbf{M}^\top)$, and $\mathbf{c}_1 \leftarrow \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$.

Proof. Let $(\mathcal{S}etup, \mathcal{S}amp\mathcal{Y}es, \mathcal{S}amp\mathcal{N}o, \mathcal{K}ey\mathcal{G}en, \mathcal{P}riv, \mathcal{P}ub)$ be a strong smooth HPS defined as Construction 3. According to Propositions 1, 2, and 3, we have

- the distributions of $\mathbf{x}_0$ and $\mathbf{c}_0$ are identical where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(\mathbf{x}_0, w) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$, and $\mathbf{c} \leftarrow \mathrm{Im}(\mathbf{M}^\top)$ ($\because$ Proposition 1).
- the distributions of $\mathbf{x}_1$ and $\mathbf{c}_1$ are identical where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(\mathbf{x}_0, w) \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$, and $\mathbf{c} \leftarrow \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ ($\because$ Proposition 2).
- it holds that for any $\mathcal{G} = \{g_\lambda\} \in \mathcal{C}_2$,

  $$|\Pr[g_\lambda(\mathsf{pp}, \mathbf{x}_0) = 1] - \Pr[g_\lambda(\mathsf{pp}, \mathbf{x}_1) = 1]| \leq \mathsf{negl}(\lambda)$$

  where $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$, $(\mathbf{x}_0, w) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$, and $\mathbf{x}_1 \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$ ($\because$ Proposition 3).

Moreover, the distribution of $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$ depends only on the distribution of $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$. Therefore, for any $\mathcal{G} = \{g_\lambda\} \in \mathcal{C}_2$, we have

$$|\Pr[g_\lambda(\mathbf{M}, \mathbf{c}_0) = 1] - \Pr[g_\lambda(\mathbf{M}, \mathbf{c}_1) = 1]| \leq \mathsf{negl}(\lambda),$$

where $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$, $\mathbf{c}_0 \leftarrow \mathrm{Im}(\mathbf{M}^\top)$, and $\mathbf{c}_1 \leftarrow \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$, completing the proof of Lemma 18. □

We now construct $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$ that distinguishes $\mathbf{c} \leftarrow \mathrm{Im}(\mathbf{M}^\top)$ and $\mathbf{c} \leftarrow \{0,1\}^\lambda \backslash \mathrm{Im}(\mathbf{M}^\top)$ with advantage $|\epsilon_{i-1} - \epsilon_i|$, which contradicts to Lemma 18.

b_λ takes as input $(\mathbf{M}, \mathbf{c})$, which is generated as $\mathbf{M} \leftarrow \mathsf{ZeroSamp}(\lambda)$ and $\mathbf{c}$ sampled as $\mathbf{c} \leftarrow \mathrm{Im}(\mathbf{M}^\top)$ or $\mathbf{c} \leftarrow \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ from its challenger. Then, it sets $ek := \mathbf{M}$ and $\mathbf{c}_{i,1} := \mathbf{c}$. Next, b_λ samples $x \leftarrow \{0,1\}^\lambda$, $(\mathbf{c}_{j,0}, \mathbf{c}_{j,1}) \leftarrow \mathrm{Im}(\mathbf{M}^\top) \times \mathrm{Im}(\mathbf{M}^\top)$ for all $j \in [i-1]$, $\mathbf{c}_{i,0} \leftarrow \mathrm{Im}(\mathbf{M}^\top)$, and $(\mathbf{c}_{j,0}, \mathbf{c}_{j,1}) \leftarrow \mathrm{Im}(\mathbf{M}^\top) \times \{0,1\}^\lambda \setminus \mathrm{Im}(\mathbf{M}^\top)$ for all $j \in \{i+1, \cdots, \lambda\}$. Next, b_λ sets $X := (x, (\mathbf{c}_{1,0}, \mathbf{c}_{1,1}), (\mathbf{c}_{2,0}, \mathbf{c}_{2,1}), \cdots, (\mathbf{c}_{\lambda,0}, \mathbf{c}_{\lambda,1}))$ and computes $Y = \mathsf{Eval}_\lambda(ek, X)$. Finally, b_λ gives (ek, Y) to a_λ. When a_λ output X^*, if $Y = \mathsf{Eval}_\lambda(ek, X^*)$, b_λ outputs 1. Otherwise, it outputs 0.

Since all operations in b_λ are performed in NC^1, we have $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$.

One can see that when $\mathbf{c} \leftarrow \mathrm{Im}(\mathbf{M}^\top)$ (resp., $\mathbf{c} \leftarrow \{0,1\}^\lambda \backslash \mathrm{Im}(\mathbf{M}^\top)$), the view of a_λ is identical to its view in **Game i-1** (resp., **Game i**), i.e., b_λ outputs 1 with probability ϵ_{i-1} (resp., ϵ_i). Therefore, $\mathcal{B} = \{b_\lambda\}$ distinguishes $\mathbf{c} \leftarrow \mathrm{Im}(\mathbf{M}^\top)$ and $\mathbf{c} \leftarrow \{0,1\}^\lambda \backslash \mathrm{Im}(\mathbf{M}^\top)$ with advantage $|\epsilon_{i-1} - \epsilon_i|$, which should be negligible according to Lemma 18, completing the proof of Lemma 17. □

Lemma 19 *If $\mathcal{A} = \{a_\lambda\}$ succeeds with advantage ϵ_λ in* **Game $\boldsymbol{\lambda}$**, *then $\epsilon_\lambda = \mathsf{negl}(\lambda)$.*

Proof. We now construct $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$ that breaks the one-wayness of $\mathcal{F} = \{f_\lambda\}$ with advantage ϵ_n.

b_λ takes as input y, which is generated as $x \leftarrow \{0,1\}^\lambda$ and $y = f_\lambda(y)$ from its challenger. Then, b_λ runs $(ek, tk) \leftarrow \mathsf{KeyGen}_\lambda$, parses $ek := \mathbf{M}$, and samples $((\mathbf{c}_{0,0}, \mathbf{c}_{0,1}), \cdots, (\mathbf{c}_{\lambda,0}, \mathbf{c}_{\lambda,1})) \leftarrow \mathrm{Im}(\mathbf{M}^\top)^{2\lambda}$. Next b_λ gives $(y, (\mathbf{c}_{0,0}, \mathbf{c}_{0,1}), \cdots, (\mathbf{c}_{\lambda,0}, \mathbf{c}_{\lambda,1}))$ to a_λ. When a_λ outputs X^*, b_λ parses $X^* := (x^*(\mathbf{c}^*_{0,0}, \mathbf{c}^*_{0,1}), \cdots, (\mathbf{c}^*_{\lambda,0}, \mathbf{c}^*_{\lambda,1}))$ and outputs x^*.

Since all operations in b_λ are performed in NC^1, we have $\mathcal{B} = \{b_\lambda\} \in \mathsf{NC}^1$.

One can see that the view of a_λ is identical to its view in **Game $\boldsymbol{\lambda}$**, i.e., b_λ outputs x^* such that $y = f_\lambda(x^*)$ with probability ϵ_λ. Therefore, $\mathcal{B} = \{b_\lambda\}$ breaks the one-wayness of $\mathcal{F} = \{f_\lambda\}$ with advantage ϵ_λ, which should be negligible, completing the proof of Lemma 19.

Since for $i \in [\lambda]$, $|\epsilon_{i-1} - \epsilon_i| = \mathsf{negl}(\lambda)$, $\epsilon_\lambda = \mathsf{negl}(\lambda)$, we have

$$\epsilon_0 \leq \sum_{i=1}^{\lambda} |\epsilon_{i-1} - \epsilon_i| + \epsilon_\lambda = \mathsf{negl}(\lambda).$$

Therefore, Construction 4 satisfies one-wayness. □

Putting all the above together, Theorem 3 immediately follows. □

6 Conclusion

In this paper, we formalize fine-grained OWPs, HPSs (which in turn derives a CCA-secure PKE), and TDFs, and show how to construct the NC^1 versions of them secure against NC^1 adversaries. Compared with traditional cryptographic primitives, our schemes treat restricted class of adversaries, while they can be run more efficiently and are only based on the mild worst case assumption $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$. It remains open how to construct more fine-grained primitives not implied by our results, such as pseudo-random functions and signature schemes, in the same model.

Acknowledgements

A part of this work was supported by NTT Secure Platform Laboratories, JST OPERA JPMJOP1612, JST CREST JPMJCR14D6, JSPS KAKENHI JP16H01705, JP17H01695, and the Sichuan Science and Technology Program under Grant 2017GZDZX0002 and 2018GZDZX0006.

References

1. Michel Abdalla, Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, and David Pointcheval. SPHF-friendly non-interactive commitments. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 214–234. Springer, Heidelberg, December 2013.
2. M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.
3. Miklós Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant depth circuits (preliminary version). In *26th Annual Symposium on Foundations of Computer Science*, pages 11–19. IEEE Computer Society Press, October 1985.
4. Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. Erratum for: on basing one-way functions on NP-hardness. In Leonard J. Schulman, editor, *42nd Annual ACM Symposium on Theory of Computing*, pages 795–796. ACM Press, June 2010.
5. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th Annual Symposium on Foundations of Computer Science*, pages 166–175. IEEE Computer Society Press, October 2004.
6. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in nc^0. *Computational Complexity*, 17(1):38–69, 2008.
7. Gilad Asharov and Gil Segev. On constructing one-way permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 512–541. Springer, Heidelberg, January 2016.
8. Yonatan Aumann, Yan Zong Ding, and Michael O. Rabin. Everlasting security in the bounded storage model. *IEEE Trans. Information Theory*, 48(6):1668–1680, 2002.
9. Yonatan Aumann and Michael O. Rabin. Information theoretically secure communication in the limited storage space model. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 65–79. Springer, Heidelberg, August 1999.
10. Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 55–72. Springer, Heidelberg, March 2008.

11. Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, November 1984.
12. Ravi B. Boppana and J. C. Lagarias. One- way functions and circuit complexity. In *Structure in Complexity Theory, Proceedings of the Conference hold at the University of California, Berkeley, California, USA, June 2-5, 1986*, pages 51–65, 1986.
13. Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 292–306. Springer, Heidelberg, August 1997.
14. Matteo Campanelli and Rosario Gennaro. Fine-grained secure computation. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 66–97. Springer, Heidelberg, November 2018.
15. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, Heidelberg, August 1998.
16. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, Heidelberg, April / May 2002.
17. Akshay Degwekar, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Fine-grained cryptography. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 533–562. Springer, Heidelberg, August 2016.
18. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.
19. Yan Zong Ding. Oblivious transfer in the bounded storage model. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 155–170. Springer, Heidelberg, August 2001.
20. Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 446–472. Springer, Heidelberg, February 2004.
21. Stefan Dziembowski and Ueli M. Maurer. On generating the initial key in the bounded-storage model. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 126–137. Springer, Heidelberg, May 2004.
22. Shohei Egashira, Yuyu Wang, and Keisuke Tanaka. Fine-grained cryptography revisited. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 637–666. Springer, Heidelberg, December 2019.
23. Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 260–270, 1981.
24. Sanjam Garg, Romain Gay, and Mohammad Hajiabadi. New techniques for efficient trapdoor functions and applications. *IACR Cryptology ePrint Archive*, 2018:872, 2018.

25. Sanjam Garg and Mohammad Hajiabadi. Trapdoor functions from the computational Diffie-Hellman assumption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 362–391. Springer, Heidelberg, August 2018.
26. Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 524–543. Springer, Heidelberg, May 2003. `http://eprint.iacr.org/2003/032.ps.gz`.
27. Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd Annual Symposium on Foundations of Computer Science*, pages 126–135. IEEE Computer Society Press, October 2001.
28. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
29. Julia Hesse, Dennis Hofheinz, and Lisa Kohl. On tightly secure non-interactive key exchange. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 65–94. Springer, Heidelberg, August 2018.
30. Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. In *30th Annual Symposium on Foundations of Computer Science*, pages 236–241. IEEE Computer Society Press, October / November 1989.
31. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science*, pages 294–304. IEEE Computer Society Press, November 2000.
32. Charanjit S. Jutla and Arnab Roy. Relatively-sound NIZKs and password-based key-exchange. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 485–503. Springer, Heidelberg, May 2012.
33. Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 78–95. Springer, Heidelberg, May 2005.
34. Jonathan Katz and Vinod Vaikuntanathan. Round-optimal password-based authenticated key exchange. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 293–310. Springer, Heidelberg, March 2011.
35. Takahiro Matsuda. On the impossibility of basing public-coin one-way permutations on trapdoor permutations. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 265–290. Springer, Heidelberg, February 2014.
36. Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, January 1992.
37. Ralph C. Merkle. Secure communications over insecure channels. *Communications of the ACM (CACM)*, 21(4):294–299, 1978.
38. Chris J. Mitchell. A storage complexity based analogue of maurer key establishment using public channels. In Colin Boyd, editor, *5th IMA International Conference on Cryptography and Coding*, volume 1025 of *Lecture Notes in Computer Science*, pages 84–93. Springer, Heidelberg, December 1995.

39. Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pages 33–43. ACM Press, May 1989.
40. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 187–196. ACM Press, May 2008.
41. A. A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):333–338, Apr 1987.
42. John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394. ACM Press, May 1990.
43. R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 77–82, New York, NY, USA, 1987. ACM.
44. Salil P. Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 61–77. Springer, Heidelberg, August 2003.
45. Emanuele Viola. On constructing parallel pseudorandom generators from one-way functions. Cryptology ePrint Archive, Report 2005/159, 2005. `http://eprint.iacr.org/2005/159`.
46. Emanuele Viola. The complexity of distributions. In *51st Annual Symposium on Foundations of Computer Science*, pages 202–211. IEEE Computer Society Press, October 2010.

Supplementary Material

A Transformation from Collection of OWPs to OWP

We now give our construction of OWP from a collection of OWPs as follows.

Construction 5 (OWP from collection of OWPs) *Let* $(\mathcal{K}ey\mathcal{G}en = \{\mathsf{KeyGen}_\lambda\}, \mathcal{E}val = \{\mathsf{Eval}_\lambda\})$ *be a collection of* $\mathcal{C}_1$*-OWPs against* $\mathcal{C}_2$ *with the key space* K_λ *and* $\mathsf{Eval}_\lambda(k, \cdot) : D_{\lambda,k} \to D_{\lambda,k}$ *for* $k \in K_\lambda$*, where it holds that the distributions* $k \leftarrow \mathsf{KeyGen}_\lambda$ *and* $k \leftarrow K_\lambda$ *are identical. Define a function family* $\mathcal{G} = \{g_\lambda : D_\lambda \to D_\lambda\}$ *as follows.*

- $D_\lambda := \bigcup_{k \in K_\lambda}(\{k\} \times D_{\lambda,k})$.
- *For* $(k, x) \in D_\lambda$, $g_\lambda(k, x) := (k, \mathsf{Eval}_\lambda(k, x))$.

Lemma 20 *If* $(\mathcal{K}ey\mathcal{G}en, \mathcal{E}val)$ *is a collection of* $\mathcal{C}_1$*-OWPs against* $\mathcal{C}_2$ *which satisfies that the distributions* $k \leftarrow \mathsf{KeyGen}_\lambda$ *and* $k \leftarrow K_\lambda$ *are identical, the function family* $\mathcal{G}$ *is a* $\mathcal{C}_1$*-OWP against* $\mathcal{C}_2$.

Proof. Since $\mathsf{Eval}_\lambda \in \mathcal{C}_1$, we have $g_\lambda \in \mathcal{C}_1$ immediately.

Permutation. Since Eval_λ is a permutation, we have

- for any $(k, x) \in D_\lambda$, $g_\lambda(k, x) \in D_\lambda$ holds since $\mathsf{Eval}_\lambda(k, x) \in D_\lambda$.
- for any $(k, x) \neq (k', x') \in D_\lambda$, $g_\lambda(k, x) = (k, \mathsf{Eval}_\lambda(k, x)) \neq (k', \mathsf{Eval}_\lambda(k', x')) = g_\lambda(k', x')$ holds, i.e., g_λ is injective, and
- for any $(k, y) \in D_\lambda$, there must exist x such that $x = \mathsf{Eval}_\lambda(k, y)$, i.e., g_λ is surjective.

Therefore, g_λ is a permutation.

One-wayness. Let $\mathcal{A} = \{a_\lambda\}$ be any adversary in $\mathcal{C}_2$ with probability ϵ. We now construct $\mathcal{B} = \{b_\lambda\} \in \mathcal{C}_2$ that breaks the one-wayness of $(\mathcal{KeyGen}, \mathcal{Eval})$.

b_λ takes as input (k, y) which is generated as $k \leftarrow \mathsf{KeyGen}_\lambda$, $x \leftarrow D_{\lambda,k}$, and $y = \mathsf{Eval}_\lambda(k, x)$. Then, b_λ gives (k, y) to a_λ. When a_λ outputs (k', x'), b_λ outputs x'.

Since the operations of b_λ only runs a_λ and returns the output of a_λ, we have $b_\lambda \in \mathcal{C}_2$. Furthermore, since the distributions $k \leftarrow \mathsf{KeyGen}_\lambda$ and $k \leftarrow K_\lambda$ are identical, a_λ outputs (k', x) such that $g_\lambda(k', x) = (k', \mathsf{Eval}_\lambda(k', x)) = (k, y)$ with probability ϵ. Therefore, b_λ outputs x such that $\mathsf{Eval}_\lambda(k, x) = y$ with probability ϵ, completing the proof of one-wayness.

Putting all the above together, Lemma 20 immediately follows.

B Perfectly Smooth and Strong Universal$_1$ for Multi-bit

We now construct a multi-bit perfectly smooth and strong universal$_1$ NC^1-HPS against NC^1 as follows.

Construction 6 (Multi-bit NC^1-HPS) *Let $\mathcal{P} = (\mathcal{Setup}, \mathcal{SampYes}, \mathcal{SampNo}, \mathcal{KeyGen}, \mathcal{Priv}, \mathcal{Pub})$ be perfectly smooth and $\frac{1}{2}$-universal$_1$ NC^1-HPS against NC^1 in Construction 3 (see Section 4) and let n be a polynomial in λ. Define $\mathcal{Setup}' = \{\mathsf{Setup}'_\lambda\}$, $\mathcal{SampYes}' = \{\mathsf{SampYes}'_\lambda\}$, $\mathcal{SampNo}' = \{\mathsf{SampNo}'_\lambda\}$, $\mathcal{KeyGen}' = \{\mathsf{KeyGen}'_\lambda\}$, $\mathcal{Priv}' = \{\mathsf{Priv}'_\lambda\}$ and $\mathcal{Pub}' = \{\mathsf{Pub}'_\lambda\}$ as follows.*

- Setup_λ*:*
  1. *Run* $\mathsf{pp} \leftarrow \mathsf{Setup}_\lambda$*.*
  2. *Parse* $\mathsf{pp} = (X_\lambda, L_\lambda, W_\lambda, R_\lambda, SK_\lambda, PK_\lambda, \Pi_\lambda, H_\lambda, \alpha_\lambda, \mathsf{aux}_\lambda)$ *and define* X'_λ, L'_λ, W'_λ, R'_λ, SK'_λ, PK'_λ, H'_λ, Π'_λ, α'_λ, *and* aux'_λ *as follows.*
     - $X'_\lambda := X_\lambda$.
     - $L'_\lambda := L_\lambda$.
     - $W'_\lambda := W_\lambda$.
     - $R'_\lambda := R_\lambda$.
     - $SK'_\lambda := SK_\lambda^{n(\lambda)}$.
     - $PK'_\lambda := PK_\lambda^{n(\lambda)}$.
     - $\Pi'_\lambda := \Pi_\lambda^{n(\lambda)} = \{0,1\}^{n(\lambda)}$.
     - $H'_\lambda(\mathbf{sk}', \mathbf{x}) := (H_\lambda(\mathbf{sk}_1, \mathbf{x}) \cdots H_\lambda(\mathbf{sk}_{n(\lambda)}, \mathbf{x}))$.
     - $\alpha'_\lambda(\mathbf{sk}') = (\alpha_\lambda(\mathbf{sk}_1) \cdots \alpha_\lambda(\mathbf{sk}_{n(\lambda)}))$.

  - $\mathsf{aux}'_\lambda := \mathsf{pp}$.

  Here, $\mathbf{sk}' = (\mathbf{sk}_1 \cdots \mathbf{sk}_{n(\lambda)}) \in SK'_\lambda$.

  3. *Set* $\mathsf{pp}' := (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, \Pi'_\lambda, H'_\lambda, \alpha'_\lambda, \mathsf{aux}'_\lambda)$ *and output* pp'.

– $\mathsf{SampYes}'_\lambda(\mathsf{pp}')$*:*

  1. *Parse* $\mathsf{pp}' := (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, \Pi'_\lambda, H'_\lambda, \alpha'_\lambda, \mathsf{aux}'_\lambda)$ *and let* $\mathsf{aux}'_\lambda = \mathsf{pp}$.
  2. *Run* $(\mathbf{x}, \mathbf{w}) \leftarrow \mathsf{SampYes}_\lambda(\mathsf{pp})$ *and output* $(\mathbf{x}, \mathbf{w})$.

– $\mathsf{SampNo}'_\lambda(\mathsf{pp}')$*:*

  1. *Parse* $\mathsf{pp}' := (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, \Pi'_\lambda, H'_\lambda, \alpha'_\lambda, \mathsf{aux}'_\lambda)$ *and let* $\mathsf{aux}'_\lambda = \mathsf{pp}$.
  2. *Run* $\mathbf{x} \leftarrow \mathsf{SampNo}_\lambda(\mathsf{pp})$ *and output* $\mathbf{x}$.

– $\mathsf{KeyGen}'_\lambda(\mathsf{pp}')$*:*

  1. *Parse* $\mathsf{pp}' := (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, \Pi'_\lambda, H'_\lambda, \alpha'_\lambda, \mathsf{aux}'_\lambda)$.
  2. *Sample* $\mathbf{sk}' \leftarrow SK'_\lambda$.
  3. *Compute* $\mathbf{pk}' := \alpha'_\lambda(\mathbf{sk}')$ *and output* $(\mathbf{pk}', \mathbf{sk}')$.

– $\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}', \mathbf{x})$*:*

  1. *Parse* $\mathsf{pp}' := (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, \Pi'_\lambda, H'_\lambda, \alpha'_\lambda, \mathsf{aux}'_\lambda)$.
  2. *Compute* $\boldsymbol{\pi} := H'_\lambda(\mathbf{sk}', \mathbf{x})$ *and output* $\boldsymbol{\pi}$.

– $\mathsf{Pub}'_\lambda(\mathsf{pp}', \mathbf{pk}', \mathbf{x}, \mathbf{w})$*:*

  1. *Parse* $\mathbf{pk}' := (\mathbf{pk}_1 \cdots \mathbf{pk}_{n(\lambda)})$.
  2. *Compute* $\boldsymbol{\pi} := (\mathsf{Pub}_\lambda(\mathbf{pk}_1, \mathbf{x}, \mathbf{w}) \cdots \mathsf{Pub}_\lambda(\mathbf{pk}_{n(\lambda)}, \mathbf{x}, \mathbf{w}))^\top$ *and output* $\boldsymbol{\pi}$.

Lemma 21 *Let* $\mathcal{P}' = (\mathcal{Setup}', \mathcal{SampYes}', \mathcal{SampNo}', \mathcal{KeyGen}', \mathcal{Priv}', \mathcal{Pub}')$ *be the construction defined as Construction 6, then,* $\mathcal{P}'$ *is perfectly smooth and strong universal*$_1$ NC^1*-HPS against* NC^1*.*

Proof. Since $\mathcal{P} \in \mathsf{NC}^1$ and there are sampling random elements and computing $n(\lambda)$ sets of algorithms in $\mathcal{P}$ which is computable in parallel, we have $\mathcal{P}' \in \mathsf{NC}^1$.

Correctness and subset membership problem follow from the fact that $\mathcal{P}$ is the perfectly smooth NC^1-HPS against NC^1.

Perfect smoothness. We now show that for any $\mathsf{pp}' \leftarrow \mathsf{Setup}'_\lambda$, the random variables $(\mathbf{x}, \mathbf{pk}', \boldsymbol{\pi})$ and $(\mathbf{x}, \mathbf{pk}', \boldsymbol{\pi}^*)$ are identical where $\mathbf{x} \leftarrow X'_\lambda \setminus L'_\lambda$, $(\mathbf{pk}', \mathbf{sk}') \leftarrow \mathsf{KeyGen}'_\lambda(\mathsf{pp}')$, and $\boldsymbol{\pi}^* \leftarrow \Pi'_\lambda$, i.e., let S_λ be a finite set $(X'_\lambda \setminus L'_\lambda \times PK'_\lambda \times \Pi'_\lambda)$, we show that for any $(\tilde{\mathbf{x}}, \widetilde{\mathbf{pk}}', \tilde{\boldsymbol{\pi}}) \in S_\lambda$, we have $\Pr[(\mathbf{x}, \mathbf{pk}', \boldsymbol{\pi}) = (\tilde{\mathbf{x}}, \widetilde{\mathbf{pk}}', \tilde{\boldsymbol{\pi}})] = \Pr[(\mathbf{x}, \mathbf{pk}', \boldsymbol{\pi}^*) = (\tilde{\mathbf{x}}, \widetilde{\mathbf{pk}}', \tilde{\boldsymbol{\pi}})]$. Since we have

$$
\begin{aligned}
&\Pr[(\mathbf{x}, \mathbf{pk}' , \boldsymbol{\pi}) = (\tilde{\mathbf{x}}, \tilde{\mathbf{pk}}', \tilde{\boldsymbol{\pi}})] \\
&\quad = \Pr\left[\begin{matrix}\mathbf{pk}' = \tilde{\mathbf{pk}}' \wedge \\ \boldsymbol{\pi} = \tilde{\boldsymbol{\pi}}\end{matrix} \middle| \mathbf{x} = \tilde{\mathbf{x}}\right] \Pr[\mathbf{x} = \tilde{\mathbf{x}}] \\
&\quad = \Pr\left[\begin{matrix}\tilde{\mathbf{pk}}' = \alpha'_\lambda(\mathbf{sk}') \wedge \\ \tilde{\boldsymbol{\pi}} = \mathsf{Priv}'_\lambda(\mathbf{sk}', \tilde{\mathbf{x}})\end{matrix}\right] \Pr[\mathbf{x} = \tilde{\mathbf{x}}] \\
&\quad = \left(\prod_{i=1}^{n(\lambda)} \Pr\left[\begin{matrix}\tilde{\mathbf{pk}}_i = \alpha_\lambda(\mathbf{sk}_i) \wedge \\ \tilde{\pi}_i = \mathsf{Priv}_\lambda(\mathbf{sk}_i, \mathbf{x})\end{matrix}\right]\right) \Pr[\mathbf{x} = \tilde{\mathbf{x}}] \\
&\quad = \left(\prod_{i=1}^{n(\lambda)} \Pr\left[\begin{matrix}\tilde{\mathbf{pk}}_i = \alpha_\lambda(\mathbf{sk}_i) \wedge \\ \bar{\pi}_i = \pi_i^*\end{matrix}\right]\right) \Pr[\mathbf{x} = \tilde{\mathbf{x}}] \quad (\because \textit{perfect smoothness of } \mathcal{P}) \\
&\quad = \Pr\left[\begin{matrix}\tilde{\mathbf{pk}}' = \alpha'_\lambda(\mathbf{sk}') \wedge \\ \boldsymbol{\pi}^* = \tilde{\boldsymbol{\pi}}\end{matrix}\right] \Pr[\mathbf{x} = \tilde{\mathbf{x}}] \\
&\quad = \Pr[(\mathbf{x}, \mathbf{pk}', \boldsymbol{\pi}^*) = (\tilde{\mathbf{x}}, \tilde{\mathbf{pk}}', \tilde{\boldsymbol{\pi}})]
\end{aligned}
$$

where $\mathbf{sk}' = (\mathbf{sk}_1 \cdots \mathbf{sk}_{n(\lambda)})$ and $\tilde{\mathbf{pk}}' = (\tilde{\mathbf{pk}}_1 \cdots \tilde{\mathbf{pk}}_{n(\lambda)})$, $\mathcal{P}'$ satisfies perfect smoothness.

Strong universality$_1$. For any $\mathsf{pp}' \leftarrow \mathsf{Setup}'_\lambda$, $\mathbf{pk}' \in PK'_\lambda$, $\mathbf{x} \in X'_\lambda \setminus L'_\lambda$, and $\boldsymbol{\pi} \in \Pi'_\lambda$, we have

$$
\begin{aligned}
\Pr[\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}', \mathbf{x}) = \boldsymbol{\pi} | \alpha'_\lambda(\mathbf{sk}') = \mathbf{pk}'] &= \prod_{i=1}^{n(\lambda)} \Pr[\mathsf{Priv}_\lambda(\mathsf{pp}', \mathbf{sk}_i, \mathbf{x}) = \pi_i | \alpha_\lambda(\mathbf{sk}_i) = \mathbf{pk}_i] \\
&\leq \prod_{i=1}^{n(\lambda)} \left(\frac{1}{2}\right) \quad (\because \frac{1}{2}\textit{-universality}_1 \textit{ of } \mathcal{P}) \\
&= \frac{1}{2^{n(\lambda)}} = \mathsf{negl}(\lambda)
\end{aligned}
$$

where $\mathbf{sk}' = (\mathbf{sk}_1 \cdots \mathbf{sk}_{n(\lambda)})$ and $\mathbf{pk}' = (\mathbf{pk}_1 \cdots \mathbf{pk}_{n(\lambda)})$, and it follows that $\mathcal{P}'$ satisfies strong universality$_1$.

This completes the proof of Lemma 21. □

C Strong Universal$_2$ for $L \times \{0,1\}^n$

Next, we give the construction of strong universal$_2$ NC^1-HPS against NC^1 such that the language L_λ supports $\{0,1\}^n$. Our construction is as follows.

Construction 7 *Let* $\mathcal{P}' = (\mathcal{S}etup', \mathcal{S}amp\mathcal{Y}es', \mathcal{S}amp\mathcal{N}o', \mathcal{K}ey\mathcal{G}en', \mathcal{P}riv', \mathcal{P}ub')$ *be strong universal*$_1$ NC^1*-HPS against* NC^1 *in Construction 6 (see Section B) and let* n *be a polynomial in* λ. *Define* $\mathcal{S}etup^\dagger = \{\mathsf{Setup}^\dagger_\lambda\}$,

$\mathcal{S}amp\mathcal{Y}es^\dagger = \{\mathsf{SampYes}^\dagger_\lambda\}$, $\mathcal{S}amp\mathcal{N}o^\dagger = \{\mathsf{SampNo}^\dagger_\lambda\}$, $\mathcal{K}ey\mathcal{G}en^\dagger = \{\mathsf{KeyGen}^\dagger_\lambda\}$, $\mathcal{P}riv^\dagger = \{\mathsf{Priv}^\dagger_\lambda\}$ *and* $\mathcal{P}ub^\dagger = \{\mathsf{Pub}^\dagger_\lambda\}$ *as follows.*

- $\mathsf{Setup}^\dagger_\lambda$:
  1. *Run* $\mathsf{pp}' \leftarrow \mathsf{Setup}'_\lambda$.
  2. *Parse* $\mathsf{pp}' = (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, \Pi'_\lambda, H'_\lambda, \alpha'_\lambda, \mathsf{aux}'_\lambda)$ *and define* $X^\dagger_\lambda$, $L^\dagger_\lambda$, $W^\dagger_\lambda$, $R^\dagger_\lambda$, $SK^\dagger_\lambda$, $PK^\dagger_\lambda$, $H^\dagger_\lambda$, $\Pi^\dagger_\lambda$, $\alpha^\dagger_\lambda$, *and* $\mathsf{aux}^\dagger_\lambda$ *as follows.*
     - $X^\dagger_\lambda := X'_\lambda \times \{0,1\}^{n(\lambda)}$.
     - $L^\dagger_\lambda := L'_\lambda \times \{0,1\}^{n(\lambda)}$.
     - $W^\dagger_\lambda := W'_\lambda$.
     - $R^\dagger_\lambda := R'_\lambda \times \{0,1\}^{n(\lambda)}$.
     - $SK^\dagger_\lambda := (SK'_\lambda)^{2(\lambda+n(\lambda))}$.
     - $PK^\dagger_\lambda := (PK'_\lambda)^{2(\lambda+n(\lambda))}$.
     - $\Pi^\dagger_\lambda := \Pi'_\lambda$.
     - $H^\dagger_\lambda(\mathbf{sk}^\dagger, (\mathbf{x}, \boldsymbol{c})) := \bigoplus_{i=1}^{\lambda+n(\lambda)} H'_\lambda(\mathbf{sk}'_{i,z_i}, \mathbf{x})$ *where* $\boldsymbol{z} = \mathbf{x}||\boldsymbol{c}$ *(let* $\mathbf{x}||\boldsymbol{c} = (\mathbf{x}^\top \ \ \boldsymbol{c}^\top)^\top$*).*
     - $\alpha^\dagger_\lambda(\mathbf{sk}^\dagger) = (\alpha'_\lambda(\mathbf{sk}'_{1,0}) \ \ \alpha'_\lambda(\mathbf{sk}'_{1,1}) \cdots \alpha'_\lambda(\mathbf{sk}'_{\lambda+n(\lambda),0}) \ \ \alpha'_\lambda(\mathbf{sk}'_{\lambda+n(\lambda),1}))$.
     - $\mathsf{aux}^\dagger_\lambda := \mathsf{pp}'_\lambda$.

     Here, $\mathbf{sk}^\dagger = (\mathbf{sk}'_{1,0} \ \ \mathbf{sk}'_{1,1} \cdots \mathbf{sk}'_{\lambda+n(\lambda),0} \ \ \mathbf{sk}'_{\lambda+n(\lambda),1}) \in SK^\dagger_\lambda$.
  3. *Set* $\mathsf{pp}^\dagger := (X^\dagger_\lambda, L^\dagger_\lambda, W^\dagger_\lambda, R^\dagger_\lambda, SK^\dagger_\lambda, PK^\dagger_\lambda, \Pi^\dagger_\lambda, H^\dagger_\lambda, \alpha^\dagger_\lambda, \mathsf{aux}^\dagger_\lambda)$ *and output* $\mathsf{pp}^\dagger$.
- $\mathsf{SampYes}^\dagger_\lambda(\mathsf{pp}^\dagger)$:
  1. *Parse* $\mathsf{pp}^\dagger = (X^\dagger_\lambda, L^\dagger_\lambda, W^\dagger_\lambda, R^\dagger_\lambda, SK^\dagger_\lambda, PK^\dagger_\lambda, \Pi^\dagger_\lambda, H^\dagger_\lambda, \alpha^\dagger_\lambda, \mathsf{aux}^\dagger_\lambda)$ *and let* $\mathsf{aux}^\dagger_\lambda = \mathsf{pp}'$.
  2. *Run* $(\mathbf{x}, \mathbf{w}) \leftarrow \mathsf{SampYes}'_\lambda(\mathsf{pp}')$ *and sample* $\boldsymbol{c} \leftarrow \{0,1\}^{n(\lambda)}$.
  3. *Output* $((\mathbf{x}, \boldsymbol{c}), \mathbf{w})$.
- $\mathsf{SampNo}^\dagger_\lambda(\mathsf{pp}^\dagger)$:
  1. *Parse* $\mathsf{pp}^\dagger = (X^\dagger_\lambda, L^\dagger_\lambda, W^\dagger_\lambda, R^\dagger_\lambda, SK^\dagger_\lambda, PK^\dagger_\lambda, \Pi^\dagger_\lambda, H^\dagger_\lambda, \alpha^\dagger_\lambda, \mathsf{aux}^\dagger_\lambda)$ *and let* $\mathsf{aux}^\dagger_\lambda = \mathsf{pp}'$.
  2. *Run* $\mathbf{x} \leftarrow \mathsf{SampNo}'_\lambda(\mathsf{pp}')$ *and sample* $\boldsymbol{c} \leftarrow \{0,1\}^{n(\lambda)}$.
  3. *Output* $(\mathbf{x}, \boldsymbol{c})$.
- $\mathsf{KeyGen}^\dagger_\lambda(\mathsf{pp}^\dagger)$:
  1. *Parse* $\mathsf{pp}^\dagger = (X^\dagger_\lambda, L^\dagger_\lambda, W^\dagger_\lambda, R^\dagger_\lambda, SK^\dagger_\lambda, PK^\dagger_\lambda, \Pi^\dagger_\lambda, H^\dagger_\lambda, \alpha^\dagger_\lambda, \mathsf{aux}^\dagger_\lambda)$.
  2. *Sample* $\mathbf{sk}^\dagger \leftarrow SK^\dagger_\lambda$.
  3. *Compute* $\mathbf{pk}^\dagger := \alpha^\dagger_\lambda(\mathbf{sk}^\dagger)$ *and output* $(\mathbf{pk}^\dagger, \mathbf{sk}^\dagger)$.
- $\mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}, \boldsymbol{c}))$:
  1. *Parse* $\mathsf{pp}^\dagger = (X^\dagger_\lambda, L^\dagger_\lambda, W^\dagger_\lambda, R^\dagger_\lambda, SK^\dagger_\lambda, PK^\dagger_\lambda, \Pi^\dagger_\lambda, H^\dagger_\lambda, \alpha^\dagger_\lambda, \mathsf{aux}^\dagger_\lambda)$.
  2. *Compute* $\boldsymbol{\pi} := H^\dagger_\lambda(\mathbf{sk}^\dagger, (\mathbf{x}, \boldsymbol{c}))$ *and output* $\boldsymbol{\pi}$.
- $\mathsf{Pub}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{pk}^\dagger, (\mathbf{x}, \boldsymbol{c}), \mathbf{w})$:
  1. *Parse* $\mathbf{pk}^\dagger = (\mathbf{pk}'_{1,0} \ \ \mathbf{pk}'_{1,1} \cdots \mathbf{pk}'_{\lambda+n(\lambda),0} \ \ \mathbf{pk}'_{\lambda+n(\lambda),1})$.
  2. *let* $\boldsymbol{z} = \mathbf{x}||\boldsymbol{c}$ *and compute* $\pi := \bigoplus_{i=1}^{\lambda+n(\lambda)} \mathsf{Pub}'_\lambda(\mathbf{pk}'_{i,z_i}, \mathbf{x}, \mathbf{w})$.
  3. *Output* $\boldsymbol{\pi}$.

Lemma 22 *Let $\mathcal{P}^\dagger = (\mathcal{S}etup^\dagger, \mathcal{S}amp\mathcal{Y}es^\dagger, \mathcal{S}amp\mathcal{N}o^\dagger, \mathcal{K}ey\mathcal{G}en^\dagger, \mathcal{P}riv^\dagger, \mathcal{P}ub^\dagger)$ be the construction defined as Construction 7, then, $\mathcal{P}^\dagger$ is strong universal$_2$ NC^1-HPS against NC^1.*

Proof. Since $\mathcal{P}' \in \mathsf{NC}^1$ and there are sampling random elements, xoring $\lambda + n(\lambda)$ strings of λ bits, and computing $\lambda + n(\lambda)$ sets of algorithms in $\mathcal{P}'$ which is computable in parallel, we have $\mathcal{P}^\dagger \in \mathsf{NC}^1$.

Correctness and subset membership problem are follows from the fact that $\mathcal{P}'$ is the strong universal$_1$ NC^1-HPS against NC^1.

Strong universality$_2$. We now show that $\mathcal{P}^\dagger$ satisfies strong universal$_2$, i.e. we have to show that for any $\mathsf{pp}^\dagger \leftarrow \mathsf{Setup}^\dagger_\lambda$, $\mathbf{pk}^\dagger \in PK^\dagger_\lambda$, $(\mathbf{x}, \boldsymbol{c}), (\mathbf{x}^*, \boldsymbol{c}^*) \in X^\dagger_\lambda$ with $(\mathbf{x}, \boldsymbol{c}) \notin L^\dagger_\lambda \cup \{(\mathbf{x}^*, \boldsymbol{c}^*)\}$ and $\boldsymbol{\pi}, \boldsymbol{\pi}^* \in \mathit{\Pi}^\dagger$, conditioned on $\boldsymbol{\pi}^* = \mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}^*, \boldsymbol{c}^*))$ and $\mathbf{pk}^\dagger = \alpha^\dagger_\lambda(\mathbf{sk}^\dagger)$, the probability that $\boldsymbol{\pi} = \mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}, \boldsymbol{c}))$ is $\mathsf{negl}(\lambda)$, where $\mathbf{sk}^\dagger = (\mathbf{sk}'_{1,0}\ \mathbf{sk}'_{1,1} \cdots \mathbf{sk}'_{\lambda+n(\lambda),0}\ \mathbf{sk}'_{\lambda+n(\lambda),1}) \leftarrow SK^\dagger_\lambda$.

Since $(\mathbf{x}, \boldsymbol{c}) \neq (\mathbf{x}^*, \boldsymbol{c}^*)$, there is $i \in [\lambda]$ such that $z_i \neq z_i^*$ where $\boldsymbol{z} = \mathbf{x}||\boldsymbol{c}$ and $\boldsymbol{z}^* = \mathbf{x}^*||\boldsymbol{c}^*$. Without loss of generality, we can assume that $i = 1$. When we fix the values of $\mathbf{sk}'_{2,0}, \mathbf{sk}'_{2,1}, \cdots, \mathbf{sk}'_{\lambda+n(\lambda),0}, \mathbf{sk}'_{\lambda+n(\lambda),1}$, the value of $\boldsymbol{\pi}$ depends only on $\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}'_{1,z_\lambda}, (\mathbf{x}, \boldsymbol{c}))$. Therefore, for all $\mathbf{pk}^\dagger = (\mathbf{pk}'_{1,0}\ \mathbf{pk}'_{1,1} \cdots \mathbf{pk}'_{\lambda+n(\lambda),0}\ \mathbf{pk}'_{\lambda+n(\lambda),1}) \in PK^\dagger_\lambda$, $(\mathbf{x}, \boldsymbol{c}), (\mathbf{x}^*, \boldsymbol{c}^*) \in X^\dagger_\lambda$, and $\boldsymbol{\pi}, \boldsymbol{\pi}^* \in \mathit{\Pi}^\dagger_\lambda$ with $(\mathbf{x}, \boldsymbol{c}) \notin L^\dagger_\lambda \cup \{(\mathbf{x}^*, \boldsymbol{c}^*)\}$, it holds that

$$\begin{aligned}
\Pr\ [\ &\mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}, \boldsymbol{c})) = \boldsymbol{\pi} \mid \mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}^*, \boldsymbol{c}^*)) = \boldsymbol{\pi}^* \wedge \alpha^\dagger_\lambda(\mathbf{sk}^\dagger) = \mathbf{pk}^\dagger] \\
&= \Pr[\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}'_{1,z_1}, \mathbf{x}) = \tilde{\boldsymbol{\pi}} \mid \mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}^*, \boldsymbol{c}^*)) = \boldsymbol{\pi}^* \wedge \alpha^\dagger_\lambda(\mathbf{sk}^\dagger) = \mathbf{pk}^\dagger] \\
&= \Pr[\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}'_{1,z_1}, \mathbf{x}) = \tilde{\boldsymbol{\pi}} \mid \alpha^\dagger_\lambda(\mathbf{sk}^\dagger) = \mathbf{pk}^\dagger] && (1) \\
&= \Pr[\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}'_{1,z_1}, \mathbf{x}) = \tilde{\boldsymbol{\pi}} \mid \alpha'_\lambda(\mathbf{sk}'_{1,z_1}) = \mathbf{pk}'_{1,\mathbf{x}_\lambda}] && (2) \\
&= \mathsf{negl}(\lambda), && (3)
\end{aligned}$$

where $\tilde{\boldsymbol{\pi}} = \boldsymbol{\pi} \oplus \left(\bigoplus_{i=2}^{\lambda+n(\lambda)} \mathsf{Pub}'_\lambda(\mathbf{pk}'_{i,z_i}, \mathbf{x}, \mathbf{w})\right)$, and it follows that $\mathcal{P}^\dagger$ satisfies strong universality$_2$.

Equation (1) follows from the fact that $\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}'_{1,z_1}, \mathbf{x})$ is independent of $\mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}^*, \boldsymbol{c}^*))$. Equation (2) is satisfied since $\mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}'_{1,z_1}, \mathbf{x})$ depends only on $\mathbf{sk}'_{1,z_1}$. Equation (3) immediately follows strong universality$_1$ of $\mathcal{P}'$.

This completes the proof of Lemma 22. □

D Fine-Grained CCA-Secure Public Key Encryption

In this section, we give the definition of NC^1-CCA-secure PKE against NC^1 circuits and give a construction by instantiating the Cramer-Shoup PKE scheme [15] with our HPS in Section 4.

D.1 Definition

Definition 9 (CCA-secure Public Key Encryption) *Let n be a polynomial in λ. Let $\mathcal{KeyGen} = \{\mathsf{KeyGen}_\lambda : \phi \to SK_\lambda \times PK_\lambda\}$, $\mathcal{Enc} = \{\mathsf{Enc}_\lambda : PK_\lambda \times \{0,1\}^{n(\lambda)} \to C_\lambda\}$, and $\mathcal{Dec} = \{\mathsf{Dec}_\lambda : SK_\lambda \times C_\lambda \to \{0,1\}^{n(\lambda)}\}$ be function families. $(\mathcal{KeyGen}, \mathcal{Enc}, \mathcal{Dec})$ is a* (k,l)-CCA-secure $\mathcal{C}_1$-public key encryption scheme (PKE) against $\mathcal{C}_2$ *if:*

- **Correctness:** *For any $\lambda \in \mathbb{N}$ and any $m \in \{0,1\}^{n(\lambda)}$:*

$$\Pr\left[\mathsf{Dec}_\lambda(\mathsf{sk}, c) = m \;\middle|\; \begin{array}{c} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_\lambda \\ c \leftarrow \mathsf{Enc}_\lambda(\mathsf{pk}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

- (k,l)**-CCA-security:** *For any $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2) = (\{g_{1,\lambda}\}, \{g_{2,\lambda}\}) \in \mathcal{C}_2$, and any $\lambda \in \mathbb{N}$:*

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{array}{c} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}_\lambda \\ b \leftarrow \{0,1\} \\ (m_1, m_2, \mathsf{aux}) \leftarrow g_{1,\lambda}^{\mathsf{Dec}_\lambda(\mathsf{sk},\cdot)}(\mathsf{pk}) \\ c \leftarrow \mathsf{Enc}_\lambda(\mathsf{pk}, m_b) \\ \tilde{b} \leftarrow g_{2,\lambda}^{\mathsf{Dec}_\lambda(\mathsf{sk},\cdot)}(c, \mathsf{aux}) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

  where $g_{1,\lambda}$ and $g_{2,\lambda}$ are allowed to do k rounds of adaptive queries to $\mathsf{Dec}_\lambda(\mathsf{sk}, \cdot)$ and each round they query l inputs.

D.2 Construction

We now give our construction of NC^1-CCA PKE against NC^1 from perfectly smooth NC^1-HPS against NC^1 and strong universal$_2$ NC^1-HPS against NC^1.

Construction 8 *Let $\mathcal{P}' = (\mathcal{Setup}', \mathcal{SampYes}', \mathcal{SampNo}', \mathcal{KeyGen}', \mathcal{Priv}', \mathcal{Pub}')$ be perfectly smooth NC^1-HPS against NC^1 defined as Construction 6 and $\mathcal{P}^\dagger = (\mathcal{Setup}^\dagger, \mathcal{SampYes}^\dagger, \mathcal{SampNo}^\dagger, \mathcal{KeyGen}^\dagger, \mathcal{Priv}^\dagger, \mathcal{Pub}^\dagger)$ be strong universal$_2$ NC^1-HPS against NC^1 defined as Construction 7. Then, define $(\mathcal{KeyGen}, \mathcal{Enc}, \mathcal{Dec})$ as follows.*

- KeyGen_λ*:*
  1. *Run* $\mathsf{pp}' := (X'_\lambda, L'_\lambda, W'_\lambda, R'_\lambda, SK'_\lambda, PK'_\lambda, H'_\lambda, \Pi'_\lambda, \alpha'_\lambda) \leftarrow \mathsf{Setup}'_\lambda$ *and* $(\mathbf{pk}', \mathbf{sk}') \leftarrow \mathsf{KeyGen}'_\lambda(\mathsf{pp}')$.
  2. *Set* $\mathsf{pp}^\dagger := (X^\dagger_\lambda, L^\dagger_\lambda, W^\dagger_\lambda, R^\dagger_\lambda, SK^\dagger_\lambda, PK^\dagger_\lambda, H^\dagger_\lambda, \Pi^\dagger_\lambda, \alpha^\dagger_\lambda)$ *based on the procedure of* $\mathsf{Setup}^\dagger_\lambda$.
  3. *Set* $\mathsf{pk} := (\mathsf{pp}', \mathsf{pp}^\dagger, \mathbf{pk}', \mathbf{pk}^\dagger)$ *and* $\mathsf{sk} := (\mathbf{sk}', \mathbf{sk}^\dagger)$.
  4. *Output* $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}_\lambda(\mathsf{pk}, m)$*:*
  1. *Parse* $\mathsf{pk} := (\mathsf{pp}', \mathsf{pp}^\dagger, \mathbf{pk}', \mathbf{pk}^\dagger)$.
  2. *Run* $(\mathbf{x}, \mathbf{w}) \leftarrow \mathsf{SampYes}'_\lambda(\mathsf{pp}')$.
  3. *Run* $\boldsymbol{\pi}' := \mathsf{Pub}'_\lambda(\mathsf{pp}', \mathbf{pk}', \mathbf{x}, \mathbf{w})$ *and compute* $\mathbf{c} := m \oplus \boldsymbol{\pi}' \in \{0,1\}^{n(\lambda)}$.

4. *Run* $\boldsymbol{\pi}^\dagger := \mathsf{Pub}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{pk}^\dagger, (\mathbf{x}, \boldsymbol{c}), \mathbf{w})$.
5. *Set* $c := (\mathbf{x}, \boldsymbol{c}, \boldsymbol{\pi}^\dagger)$ *and output* c.

– $\mathsf{Dec}_\lambda(\mathsf{sk}, c)$*:*
  1. *Parse* $\mathsf{sk} := (\mathbf{sk}', \mathbf{sk}^\dagger)$ *and* $c := (\mathbf{x}, \boldsymbol{c}, \boldsymbol{\pi}^\dagger)$.
  2. *Run* $\tilde{\boldsymbol{\pi}}^\dagger := \mathsf{Priv}^\dagger_\lambda(\mathsf{pp}^\dagger, \mathbf{sk}^\dagger, (\mathbf{x}, \boldsymbol{c}))$.
  3. *If* $\boldsymbol{\pi}^\dagger \neq \tilde{\boldsymbol{\pi}}^\dagger$*, then output* $\perp$ *and halt.*
  4. *Run* $\tilde{\boldsymbol{\pi}}' := \mathsf{Priv}'_\lambda(\mathsf{pp}', \mathbf{sk}', \mathbf{x})$ *and compute* $\tilde{m} := \boldsymbol{c} \oplus \tilde{\boldsymbol{\pi}}'$.
  5. *Output* $\tilde{m}$.

Theorem 4 *Let k be a constant and l be a polynomial in λ. If* $\mathsf{NC}^1 \subsetneq \oplus\mathsf{L}/\mathsf{poly}$*, then* $(\mathcal{K}ey\mathcal{G}en, \mathcal{E}nc, \mathcal{D}ec)$ *defined as Construction 6 is a (k,l)-CCA-secure* NC^1*-PKE scheme against* NC^1.

We note that the algorithms $\mathcal{K}ey\mathcal{G}en$, $\mathcal{E}nc$, and $\mathcal{D}ec$ compute XOR, equality comparison, and run $\mathcal{P}'$ and $\mathcal{P}^\dagger$(see Section B and C for $\mathcal{P}'$ and $\mathcal{P}^\dagger$). Since all these operations can be performed in NC^1, we have $\mathcal{K}ey\mathcal{G}en, \mathcal{E}nc, \mathcal{D}ec \in \mathsf{NC}^1$.

The proofs of correctness and CCA security are the same as the proofs by Cramer and Shoup [15], and we thus omit the details. Note that the adversary makes constant rounds of queries, and in each round, the simulation for queries made by the reduction algorithm is performed in NC^1. Moreover, one can see that the rest operations in the reduction algorithm can also be performed in NC^1. Therefore, Construction 8 is a CCA-secure NC^1-PKE scheme against NC^1, completing the proof of Theorem 4.

Table of Contents