Deniable Cryptosystems: Simpler Constructions and Achieving Leakage Resilience

Zhiyuan An[1,2], Haibo Tian[1,2], Chao Chen[1,2], and Fangguo Zhang[1,2](✉)

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
anzhy@mail2.sysu.edu.cn,tianhb@mail.sysu.edu.cn,chench533@mail2.sysu.edu.cn, isszhfg@mail.sysu.edu.cn

[2] Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China

Abstract. Deniable encryption (Canetti et al. in CRYPTO '97) is an intriguing primitive, which provides security guarantee against coercion by allowing a sender to convincingly open the ciphertext into a fake message. Despite the notable result by Sahai and Waters in STOC '14 and other efforts in functionality extension, all the deniable public key encryption (DPKE) schemes suffer from intolerable overhead due to the heavy building blocks, e.g., translucent sets or indistinguishability obfuscation. Besides, none of them considers the possible damage from leakage in the real world, obstructing these protocols from practical use.

To fill the gap, in this work we first present a simple and generic approach of sender-DPKE from ciphertext-simulatable encryption, which can be instantiated with nearly all the common PKE schemes. The core of this design is a newly-designed framework for flipping a bit-string that offers inverse polynomial distinguishability. Then we theoretically expound and experimentally show how classic side-channel attacks (timing or simple power attacks), can help the coercer to break deniability, along with feasible countermeasures.

Keywords: Deniable encryption · Simulatable encryption · Side-channel attacks · Leakage resilience.

1 Introduction

Deniable Encryption, firstly introduced by Canetti et al. [7], is a seemingly contradictory primitive which allows a coerced user to produce fake (but valid-looking) random coins that could open the original ciphertext to another message. More detailedly, there is an additional fake algorithm, which on inputting the original plaintext m, used randomness r, and any fake message m^*, returns some fake coins r^*. In this way, the sender can claim the questioned ciphertext to be the encryption of m^* under r^*, and the coercer can not detect the lie.

Compared with the traditional encryption notions that provide security against only passive attacks, deniable encryption provides more shields since it is coercion resistant and non-committing in the context of active attacks. In this sense,

deniable public key encryption (DPKE) can be deployed in systems where strong privacy-preserving is required, e.g., electronic voting [13], uncoercible multiparty computation [7,22], cloud storage service [12] and searchable encryption [28].

Prior Works on Deniability. Over the last decades, many approaches were proposed to build deniable encryption. The seminal work [7] provided two schemes for bit encryption using a well-defined primitive called translucent sets (TS). Following this blueprint, O'Neill et al. [33] explored non-interactive bi-deniable encryption under the weak model where both sides can fake simultaneously, along with constructions from lattice-based bi-TS. A notable breakthrough was achieved by Sahai and Waters [34], where they presented the first and only known construction supporting negligible detection probability by use of indistinguishability obfuscation ($i\mathcal{O}$) [21,24] and puncturable PRFs [4]. Recently, Agrawal et al. [2] tackled deniability by equipping fully homomorphic encryption (FHE) [23] (e.g., the BGV scheme [5]) with biased decryption. As extensions of DPKE, Gao et al. [20] studied the stronger notation of CCA-secure DPKE and provided an instantiation from extended hash proof systems; Caro et al. [10,11] built deniable function encryption by combining $i\mathcal{O}$ and delayed trapdoor circuit; Besides, Coladangelo et al. [15] explored the possible quantum setting where the encryption program is implemented under quantum circuits, and gave efficient constructions from LWE. There has also been work on fully interactive DPKE [8], where negligible bi-deniability was achieved based on $i\mathcal{O}$ and OWFs.

Current Limitations of DPKE. Although the aforementioned works settled the issue in various aspects, they all bear somewhat heavy building blocks, e.g., TS-based schemes [3,7,33] only support bit-encryption; FHE-derived one [2] has the runtime of encryption being linear of both the inverse detection probability and the size of message space; the only scheme with negligible detection probability [34] is built on the powerful $i\mathcal{O}$ which however requires sub-exponential assumptions and huge storage cost. These facts make them fall short of being deployment-friendly, let alone integrate with other cryptosystems into synthetical programs. Therefore, there has still been a challenging gap between theoretical prototypes and pragmatic systems on *deniability*. In other words, it is more desirable to construct deniable encryption from handy methods and with as practical as possible overhead (ciphertext size or runtime).

On the other side, there have been lots of work paying close attention to another security notion of PKE. Namely, the resilience to the leakage from physical hardware that encapsulates the related algorithms, e.g., side-channel attacks (SCA) from timing or power analysis [1,18,19,26,27], which are common threats to the cryptographic applications in real-world [6,31]. Previous works have also provided various manners to avoid such leakage including general models or specific countermeasures. However, there has been no headway yet that sheds light on the potential damage to deniable schemes. That is, we have no idea that, with some available side-channel information of programs where the sender operates, can the coercer distinguish the claimed randomness from the real ones, so as to breach the *deniability* of the target system? Thus, towards the practical use of DPKE, it is encouraging to explore *deniability* in the context of SCA.

Our Contributions. This work addresses the above two limitations of existing deniable encryption schemes. Our contributions are summarized in the following.

- We propose a generic construction of DPKE from ciphertext-simulatable PKE, an underpinning that can be instantiated with nearly all the common PKE schemes. In particular, we devise a subtle bit-flipping framework within a bit string to support inverse polynomial detection probability.
- We formalize the SCA-equipped coercion model for timing and simple power attacks, under which we show how deniability can be breached, as well as provide suitable countermeasures, we then evidence these results by performing relative experiments.

Scheme	Methods	Mess. space	Deniability	SCA	Cipher. size	Runtime
[7]	TS	$\{0,1\}$	$\mathcal{O}(\frac{1}{\lambda})$	✗	$\mathcal{O}(\lambda \cdot \tau_l)$	$\mathcal{O}(\lambda \cdot \tau_t)$
[34]	$i\mathcal{O}$ + PKE	$\{0,1\}$	$\mathsf{negl}(\lambda)$	✗	$\mathcal{O}(\tau_l)$	$\mathcal{O}(\tau_t)$
[2]	FHE	$\mathsf{poly}(\lambda)$	$\mathcal{O}(\frac{1}{\lambda})$	✗	$\mathcal{O}(\tau_l)$	$\tau_t \cdot \mathsf{poly}(\lambda)$
Ours	*CS*-PKE	2^λ	$\mathcal{O}(\frac{1}{\lambda})$	Against	$\mathcal{O}(\lambda \cdot \tau_l)$	$\mathcal{O}(\lambda \cdot \tau_t)$

Table 1. Comparison between known schemes and ours.

For security parameter λ, Table 1 gives an overall comparison of some known sender-DPKE and ours, where τ_l and τ_t denote the element size and runtime of the underlying methods, respectively, e.g., the ciphertext size and en/decryption runtime of the PKE used in [34]. As we will expound in Sec. 2, nearly all the common PKE schemes (e.g., ElGamal, Cramer-Shoup, Kyber) are inherently ciphertext-simulatable (*CS*), which demonstrates the superiority of our scheme in availability. Besides, the notation 2^λ in the third column means that our scheme supports the inherent message space of the used PKE scheme, while [7,34] only admits encryption of bit under whatever methods, and [2] has encryption runtime being linearly dependent of the message space. Finally, our scheme for the first time considers the issue of SCA, along with some basic countermeasures, which is a fundamental guidance towards practical applications of DPKE.

Overview Of Our Techniques. In the following, we provide more technical details of our contributions.

Generic Approach of DPKE. We begin with informally defining the *CS*-PKE, where an oblivious algorithm OEnc samples a random ciphertext ct_r relative to a public key using some randomness r, without knowing the corresponding plaintext. Its inverting algorithm IEnc, on inputting the original message and encryption randomness, simulates the above process by returning a simulated randomness r^*. Our core idea is to utilize the ability of interpreting an encryption as a randomly sampled one in *CS*-PKE to deceive the coercer. In this sense, we have to make sure that the receiver can distinguish between these two types of ciphertexts. Thus, we tag every ciphertext with an OWF ($\mathcal{H}$) value. Namely, the encryption of a message m is a pair $(\mathsf{Enc}(\mathsf{pk}, m||u), \mathcal{H}(u))$, while the oblivious

sample is $(\mathsf{ct}_r, \mathcal{H}(u))$, where u is a random nonce.

Then we give an abstract of the newly-designed bit-flipping framework. The main layout is that the encryption of a message $\boldsymbol{m}$ contains n sub-ciphertexts $\{\mathsf{ct}_i\}$ binding to the pattern of a random bit-string $\boldsymbol{s} \in \{0,1\}^n$. In particular, for $\boldsymbol{s}[i] = 0$, generate an obliviously sampled pair

$$\boldsymbol{c}_i := (\boldsymbol{c}_i^{(1)}, \boldsymbol{c}_i^{(2)}) \leftarrow (\mathsf{OEnc}(\mathsf{pk}; r_i), \mathcal{H}(u_i)).$$

while for $\boldsymbol{s}[i] = 1$, produce an honest encryption of message $\boldsymbol{m}_i$ as

$$\boldsymbol{c}_i := (\boldsymbol{c}_i^{(1)}, \boldsymbol{c}_i^{(2)}) \leftarrow (\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}||u_i; r_i), \mathcal{H}(u_i)),$$

where $\boldsymbol{m}_i$ is random over the valid message space, except for one random index $t = \mathsf{select}(\boldsymbol{s}; \boldsymbol{v})$ where select is a publicly random map into the "1"-set of the input string and $\boldsymbol{v}$ is an auxiliary nonce, $\boldsymbol{m}_t$ is the real message $\boldsymbol{m}$. The final ciphertext ct for $\boldsymbol{m}$ is $(\{\mathsf{ct}_i\}, \boldsymbol{v})$. In this way, the receiver first decodes $\{\mathsf{ct}_i\}$ in sequence to recover $\boldsymbol{s}$, i.e., set $\boldsymbol{s}[i] = 1$ iff $\boldsymbol{c}_i^{(2)}$ is the OWF image of u_i, which is decrypted from $\boldsymbol{c}_i^{(1)}$; then locates the index $t = \mathsf{select}(\boldsymbol{s}; \boldsymbol{v})$ to obtain the real message $\boldsymbol{m}_t = \boldsymbol{m}$. The negligible decryption error comes from that for "0"-mode pair, u_i will not be the preimage of $\boldsymbol{c}_i^{(2)}$ due to the *one-wayness* of $\mathcal{H}$.

To fake, the sender first samples from $\mathsf{IEnc}(\mathsf{pk}, \boldsymbol{m}, r_t)$ a simulated randomness r_t^*, which cloaks $\boldsymbol{c}_t^{(1)}$ as a random sample from $\mathsf{OEnc}(\mathsf{pk}; r_t^*)$. Then she/he flips s_t from 1 to 0 to output a faking $\boldsymbol{s}^*$ and provides all the other original randomness $\{\boldsymbol{m}_i, r_i, u_i\}$. In this way, the sender can explain ct as the encryption of $\boldsymbol{m}_{t^*}$ for $t^* = \mathsf{select}(\boldsymbol{s}^*; \boldsymbol{v})$. Further, the detection probability of a coercer is scaled by the statistical difference of $\boldsymbol{s}$ and $\boldsymbol{s}^*$, which is essentially the distance between random and one-bit-flipping sampling of a bit-string. We step forward to prove it is bounded by an inverse polynomial $\frac{1}{\sqrt{n}}$ in Thm. 1, thus our scheme shares the same security level of known schemes [2, 7, 9] from standard assumptions.

SCA *to Deniability.* We mainly consider the basic types of SCA (timing attacks [26] and simple power analysis [27]). The failure of deniability is based on a theoretical observation: there is an inherent disparity between fake opening and honesty of all the known schemes, e.g., the ways of sampling used randomness, the count of times that a subprogram is invoked. This disparity will result in the difference in operating time or power consumption within the encryption program. Then a coercer can first record such side-channel information during the execution of encryption, and demand the sender to rerun the encryption under the claimed randomness and plaintext, then detect the lie if the records of two operations have a significant change. Formally, we model the behaviors of a coercer as two steps: (passively) monitor to collect the target ciphertext ct and its SC information; (actively) coerce to obtain the internal plaintext and randomness, along with the fresh SC information.

Under the above enhanced coercion model, we examine most known DPKE schemes and ours, demonstrating that a *denial* of the original message can always be distinguished with polynomial overhead. To further evidence these theoretical

conclusions, we instantiate the schemes of [7] and ours with ElGamal encryption, then compare the consumption of CPU cycles between the honest encryption and fake opening, the experiment results (Fig. 3) show the gap is stable and effective (mostly > 54.6 µs). Finally, we provide some countermeasures to such SCA, i.e., we make encryption algorithm conduct some tiny redundancy operations, such that honesty and faking execute the very same instruction stream. Simulations on these updated schemes exhibit that now the variation of time/power consumption is changed to small than 10 ns (Fig. 4), meaning the fixing is indeed feasible. In summary, our work mounts the practical security of deniable encryption when applied in real-world systems.

Organization. In the forthcoming sections, we first recall some necessary preliminaries in Sec. 2. Then we provide a generic approach of DPKE from ciphertext-simulatable PKE in Sec. 3. Finally Sec. 4 depicts how to break deniability of known schemes and ours under the SCA-enhanced coercion model, together with suitable countermeasures.

2 Preliminaries

In this section, we define the notation and preliminaries required in this work.

Notations. Let λ denote the security parameter throughout the paper. Function $f(\lambda)$ is said to be negligible if it is $\mathcal{O}(\lambda^{-c})$ for all $c > 0$, and use $\mathsf{negl}(\lambda)$ to denote such a function of λ. $f(\lambda)$ is said to be polynomial if it is $\mathcal{O}(\lambda^{-c})$ for some constant $c > 0$, and use $\mathsf{ploy}(\lambda)$ to denote such a function of λ. Event X is said to occur with overwhelming probability in λ if $\Pr[\mathrm{X}] = 1 - \mathsf{negl}(\lambda)$. Let $\mathcal{F}(x; r)$ denote a randomized algorithm $\mathcal{F}$ runs on input x and randomness r.

Use $[n]$ to denote the integer set $\{1, \ldots, n\}$. Use bold lower-case letters (e.g., $\boldsymbol{s}$) to denote a bit-string. For $\boldsymbol{s}$, denote its i-th element as $\boldsymbol{s}[i]$, the index of its $(j+1)$-th "1" as $\mathsf{L}(\boldsymbol{s}, j)$, its hamming weight as $\mathsf{w}(\boldsymbol{s})$, its decimal as $\mathsf{dec}(\boldsymbol{s})$, its 0 and 1-index sets as $\mathcal{S}_0$ and $\mathcal{S}_1$, respectively. For a finite set $\mathcal{X}$, denote by $x \leftarrow \mathcal{X}$ sampling x uniformly from $\mathcal{X}$, and by $y \leftarrow D$ sampling y according to the distribution D. The statistical Distance between two distributions D_1 and D_2 over $\mathcal{X}$ is $\mathsf{SD}(D_1, D_2) = \frac{1}{2} \sum_{x \in \mathcal{S}} |D_1(x) - D_2(x)|$.

2.1 Sender-Deniable Public Key Encryption

We first recall the model of sender-deniable public key encryption introduced in [7], such a scheme $\mathcal{DE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake})$ has the following syntax:

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$: With the security parameter λ, generate the public and secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}; r)$: On inputting the public key pk and a message $\boldsymbol{m}$, use randomness r to produce a ciphertext ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: On inputting the secret key sk and a ciphertext ct, output a message $\boldsymbol{m}$ or $\perp$.

- $\mathsf{Fake}(\mathsf{pk}, \boldsymbol{m}, r, \boldsymbol{m}^*)$: On inputting the public key pk, original message $\boldsymbol{m}$, randomness r, and a fake message $\boldsymbol{m}^*$, output a fake randomness r^*.

Correctness. $\mathcal{DE}$ is correct if, for any security parameter λ, message $\boldsymbol{m}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, it holds that $\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}; r)) = \boldsymbol{m}] = 1 - \mathsf{negl}(\lambda)$.

Definition 1 (IND-CPA). *$\mathcal{DE}$ is IND-CPA secure if for all* PPT *adversary $\mathcal{A}$, the absolute difference of probability of outputting 1 between experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{CPA}-0}$ and $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{CPA}-1}$ is negligible.*

Experiment: $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{CPA}-b}(\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$.
$(\boldsymbol{m_0}, \boldsymbol{m_1}, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk})$.
Compute $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}_b; r)$, return ct to $\mathcal{A}$.
$b' \leftarrow \mathcal{A}_2(\mathsf{pk}, \mathsf{ct}, \mathsf{st})$. Output b'.

Definition 2 (Deniability). *$\mathcal{DE}$ satisfies deniability if for any* PPT *adversary $\mathcal{A}$, the absolute difference of probability of outputting 1 between experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-0}$ and $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-1}$ is negligible.*

Experiment: $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-b}(\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$.
$(\boldsymbol{m}, \boldsymbol{m^*}, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pk})$.
Sample r and $r^* \leftarrow \mathsf{Fake}(\mathsf{pk}, \boldsymbol{m}, r, \boldsymbol{m}^*)$.
If $b = 0$: return $D_0 = (\boldsymbol{m}^*, r, \mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}^*; r))$ to $\mathcal{A}$.
Else if $b = 1$: return $D_1 = (\boldsymbol{m}^*, r^*, \mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}; r))$ to $\mathcal{A}$.
$b' \leftarrow \mathcal{A}_2(\mathsf{pk}, D_b, \mathsf{st})$. Output b'.

2.2 Ciphertext-Simulatable Public Key Encryption

Ciphertext-simulatable PKE is a relaxed version of simulatable PKE [16], in the sense that 1) it only admits the oblivious sampling of ciphertexts; 2) the corresponding inverting algorithm additionally takes the encryption-used plaintext and randomness as input to return a randomness relative to oblivious sampling.

Formally, such PKE consists of universal algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, augmented with $(\mathsf{OEnc}, \mathsf{IEnc})$ for obliviously sampling and inverting ciphertexts:

- $\mathsf{OEnc}(\mathsf{pk}; r)$: On inputting the public key pk, use randomness r_o to sample a ciphertext ct.
- $\mathsf{IEnc}(\mathsf{pk}, m, r_e)$: On inputting the public key pk, message $\boldsymbol{m}$, randomness r_e used in the original encryption, output a randomness r_o^*.

Definition 3 (Ciphertext-Simulatability [14, 16]). *For CS-*PKE*, it holds that for all PPT distinguisher $\mathcal{D}$, message m, public key $\mathsf{pk} \leftarrow \mathsf{KGen}(1^\lambda)$,*

$$\Big| \Pr[\mathcal{D}(\mathsf{Enc}(\mathsf{pk}, m; r_e), \mathsf{IEnc}(\mathsf{pk}, m, r_e)) = 1] - \Pr[\mathcal{D}(\mathsf{OEnc}(\mathsf{pk}; r_o), r_o) = 1] \Big| \leq \mathsf{negl}(\lambda).$$

As noted in [14, 33], ciphertext-simulatability implies IND-CPA. Besides, the ongoing works [14, 16, 17, 29, 33] have shown that simulatable encryption can

be realized from nearly all the standard cryptographic assumptions, e.g., DDH (ElGamal and Cramer-Shoup), RSA (PKE from RSA-based trapdoor permutations), as well as worst-case lattice assumptions (LWE-based encryptions), these results also apply to ciphertext-simulatable PKE as it is a weaker variant.

3 Generic Construction of DPKE

In this section, we give the generic approach of DPKE from any ciphertext-simulatable PKE scheme. The sketchy roadmap is: first sample a uniform random bit-string $\boldsymbol{s}$, then use another randomness $\boldsymbol{v}$ to select a random index t of "1" in $\boldsymbol{s}$ and encrypt $\boldsymbol{m}$ at t. For $i \in \mathcal{S}_1 \setminus \{t\}$, encrypt a random message $\boldsymbol{m}_i$; otherwise, obliviously sample a random cipher. In particular, all the encryptions are operated on the message plus a random tag, whose evaluation of an OWF is also dispatched. In this way, the receiver could decrypt all the n ciphers to reassemble $\boldsymbol{s}$, so as to locate the index t binding to $\boldsymbol{m}$. To fake, the sender flips s_t to obtain a fake string $\boldsymbol{s}^*$ and index t^*, then invert-samples $\boldsymbol{c}_t$ as a random pair, so to interpret the ciphertext as the encryption of the fake message $\boldsymbol{m}_{t^*}$.

The faking probability of this design mainly hinges on the statistical distance between $\boldsymbol{s}$ and $\boldsymbol{s}^*$. Thus, below we first clarify how flipping one bit influences the randomness of a string, then give the description and analysis of our scheme.

3.1 Warm-up: Bit Flipping

We consider the issue of the remaining randomness of a made string from flipping a "1" of a random string. Specifically, we prove that $\boldsymbol{s}$ and $\boldsymbol{s}^*$ are within an inverse polynomial distance, as the following theorem shows.

Theorem 1 (Randomness of bit-flipping). *Given two distributions U and F for a bit-string, the first is the uniformly random sampling from the finite set $\mathcal{S} = \{0,1\}^n$, and the latter is the flipping case where it first samples $\boldsymbol{s}$ from $\mathcal{S}$, if $\boldsymbol{s} = 0^n$, outputs $\perp$; else it outputs a string from randomly flipping one bit in $\boldsymbol{s}$ from 1 to 0. The statistical distance between U and F is $\Theta(\frac{1}{\sqrt{n}})$.*

Proof. W.l.o.g, assume $n = 2m+1$. Consider the count k of 1 of $\boldsymbol{s}$, i.e., $\sum_{i=1}^{n} \boldsymbol{x}[i] = k$ for $k \in [0, n]$, then the probability of $\boldsymbol{s}$ for each k in R is $\frac{1}{2^n} \cdot \binom{n}{k}$; The probability $F(\boldsymbol{s})$ is more complicated. Observe that $\boldsymbol{s}$ must be obtained by flipping a "1" (indexed as i) of a string $\boldsymbol{s}'$ from $\mathcal{S}$ whose count of "1" is $k+1$. Thus, there are $n-k$ possible $\boldsymbol{s}'$ when fixing $\boldsymbol{s}$. Further, the probability of flipping $\boldsymbol{s}[i]$ in $\boldsymbol{s}'$ is $\frac{1}{k+1}$. Therefore, we get $F(\boldsymbol{s}) = \frac{1}{2^n} \cdot \binom{n}{k} \frac{n-k}{k+1}$ and the following equation:

$$
\begin{aligned}
\mathsf{SD}(U, F) &= \frac{1}{2} \cdot \sum_{\boldsymbol{x} \in \mathcal{S}} |U(\boldsymbol{x}) - F(\boldsymbol{x})| + \frac{1}{2} \cdot F(\perp) \\
&= \frac{1}{2} \cdot \sum_{k=0}^{n} \left| \frac{1}{2^n} \binom{n}{k} \left(1 - \frac{n-k}{k+1}\right) \right| + \frac{1}{2^{n+1}} \\
&= \frac{1}{2^{n+1}} \cdot \left(\sum_{k=m+1}^{n} \binom{n}{k} \left(1 - \frac{n-k}{k+1}\right) + \sum_{k=0}^{m} \binom{n}{k} \left(\frac{n-k}{k+1} - 1\right) + 1 \right).
\end{aligned}
$$

Note that $\frac{n-k}{k+1} = 1$ for $k = m$ and $\binom{n}{k} = \binom{n}{n-k}$, thus the above equation can be further simplified into

$$\begin{aligned}\mathsf{SD}(U,F) &= \frac{1}{2^{n+1}} \cdot \left(\sum_{k=0}^{m} \binom{n}{k} \left(\frac{n-k}{k+1} - \frac{k}{n-k+1} \right) + 1 \right) \\ &= \frac{1}{2^{n+1}} \cdot \left(n + \left(\sum_{k=1}^{m} \binom{n}{k+1} - \binom{n}{k-1} \right) + 1 \right) \\ &= \frac{1}{2^{n}} \cdot \binom{n}{m}.\end{aligned} \tag{1}$$

By applying Stirling's approximation, we obtain $\mathsf{SD}(U,F) \approx \frac{1}{\sqrt{\pi n}} = \Theta(\frac{1}{\sqrt{n}})$. □

In App. A, we further prove the optimality of the above one-bit flipping case, i.e., it reserves the most randomness of $\boldsymbol{s}$ under all the possible flipping manners.

3.2 The New Framework

The underlying methods are an OWF $\mathcal{H} : \mathcal{U} \to \{0,1\}^{\ell_t}$ for $\mathcal{U} = \{0,1\}^{\ell_h}$, and a ciphertext-simulatable PKE $\mathcal{E}$ with message space $\mathcal{M}' = \{0,1\}^{\ell_{m'}}$ where $\ell_{m'} = \ell_m + \ell_h$, randomness space $\mathcal{R}_e \subset \{0,1\}^{\ell_e}$ and $\mathcal{R}_o \subset \{0,1\}^{\ell_o}$ for encryption and oblivious sampling, respectively, w.l.o.g., we assume $\mathcal{R}_o = \mathcal{R}_e = \mathcal{R}$. For ease of notation, we suppress the polynomial dependence on λ of the associated parameters. Our framework of DPKE $\mathcal{DE}$ for message space $\mathcal{M} = \{0,1\}^{\ell_m}$ is as follows:

- $\mathsf{KGen}(1^\lambda)$: Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{E}.\mathsf{KGen}(1^\lambda)$, and output $\mathsf{dpk} := \mathsf{pk}$, $\mathsf{dsk} := \mathsf{sk}$.
- $\mathsf{Enc}(\mathsf{dpk}, \boldsymbol{m})$: Upon inputting dpk and $\boldsymbol{m} \in \mathcal{M}$, conduct the following:
  1. Sample $\boldsymbol{s}, \boldsymbol{v} \leftarrow \{0,1\}^n$. Abort if $\boldsymbol{s} = 0^n$; Else, determine the index $t = \mathsf{L}(\boldsymbol{s},\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}))$.
  2. For $i \in [n]$: sample $r_i \leftarrow \mathcal{R}, u_i \leftarrow \mathcal{U}$; further if $i \in \mathcal{S}_1$, sample $\boldsymbol{m}_i \leftarrow \mathcal{M}$, except that take $\boldsymbol{m}_t = \boldsymbol{m}$ for $i = t$. The internal randomness is
  $$\mathsf{Rand} := (\boldsymbol{s}, \{\boldsymbol{m}_i\}_{i \in \mathcal{S}_1 \setminus \{t\}}, \{r_i, u_i\}_{i \in [n]}).$$
  3. Finally, generate n ciphertexts $\{\boldsymbol{c}_i\}_{i \in [n]}$ under the pattern of $\boldsymbol{s}$:
     ① If $i \in \mathcal{S}_0$, set the masking ciphertext as
     $$\boldsymbol{c}_i := (\boldsymbol{c}_i^{(1)}, \boldsymbol{c}_i^{(2)}) \leftarrow (\mathcal{E}.\mathsf{OEnc}.(\mathsf{pk}; r_i), \mathcal{H}(u_i)). \tag{1}$$
     ② Else, produce the real encryption of $\boldsymbol{m}_i$ as
     $$\boldsymbol{c}_i := (\boldsymbol{c}_i^{(1)}, \boldsymbol{c}_i^{(2)}) \leftarrow (\mathcal{E}.\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}_i || u_i; r_i), \mathcal{H}(u_i)). \tag{2}$$
  4. Output $\mathsf{dct} := (\{\boldsymbol{c}_i\}_{i \in [n]}, \boldsymbol{v})$.
- $\mathsf{Dec}(\mathsf{dsk}, \mathsf{dct})$: Parse dct as $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{v})$, for $i \in [n]$, do the following:
  1. Run $\overline{\boldsymbol{m}}_i := \mathcal{E}.\mathsf{Dec}(\mathsf{dsk}, \boldsymbol{c}_i^{(1)})$. If $\overline{\boldsymbol{m}}_i = \perp$, set $\boldsymbol{e}[i] = 0$ and move to $i := i+1$;

2. Parse $\overline{\boldsymbol{m}}_i$ as $\boldsymbol{m}_i||u_i$, set $\boldsymbol{e}[i] = 1$ if $\mathcal{H}(u_i) \stackrel{?}{=} \boldsymbol{c}_i^{(2)}$, or 0 otherwise.

Output $\perp$ if $\mathsf{w}(\boldsymbol{e}) = 0$. Else, compute $t_{\boldsymbol{e}} = \mathsf{L}(\boldsymbol{e},\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{e}))$, and output $\boldsymbol{m} := \boldsymbol{m}_{t_{\boldsymbol{e}}}$.

- $\mathsf{Fake}(\mathsf{dpk}, \boldsymbol{m}, \mathsf{Rand}, \boldsymbol{m}^*)$: Upon inputting the public key dpk, real message $\boldsymbol{m}$ and used randomness Rand, along with the fake message $\boldsymbol{m}^* \in \{\boldsymbol{m}_i\}_{i\in\mathcal{S}_1\setminus\{t\}}$, conduct as follows to produce a fake randomness Rand^*:
  1. If $\boldsymbol{m}^* = \boldsymbol{m}$, output $\mathsf{Rand}^* = \mathsf{Rand}$.
  2. Else, set $\boldsymbol{s}^* = (\ldots \boldsymbol{s}[t-1]\ 0\ \boldsymbol{s}[t+1]\ldots)$ and $t^* = \mathsf{L}(\boldsymbol{s}^*,\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}^*))$.
  3. For $i \in [n]$: if $i = t$, generate $r_i^* := \mathcal{E}.\mathsf{IEnc}(\mathsf{pk}, \boldsymbol{m}, r_i)$ and set $u_i^* = u_i$; else, set $r_i^* = r_i$ and $u_i^* = u_i$, additionally set $\boldsymbol{m}_i^* = \boldsymbol{m}_i$ if $i \in \mathcal{S}_1 \setminus \{t^*\}$.
  4. Return $\mathsf{Rand}^* = (\boldsymbol{s}^*, \{\boldsymbol{m}_i^*\}_{i\in\mathcal{S}_1\setminus\{t,t^*\}}, \{r_i^*, u_i^*\}_{i\in[n]})$.

Remark 1. The above scheme is pre-planning, in the sense that the sender must choose the fake message $\boldsymbol{m}_{t^*}$ at the beginning of the encryption.

Theorem 2. *Suppose that $\mathcal{E}$ is correct and $\mathcal{H}$ is one-way, then $\mathcal{DE}$ is correct.*

Proof. We prove the correctness of $\mathcal{DE}$ by showing that the recovered $\boldsymbol{e}$ in Dec is the exact $\boldsymbol{s}$ used in Enc. Note that for any honestly generated ciphertext $\mathsf{dct} : \{\boldsymbol{c}_i\}$, it holds that for $i \in [n]$:

1. If $i \in \mathcal{S}_1$, $\boldsymbol{c}_i$ is produced as Eq. (2), the honest encryption of $\boldsymbol{m}_i$, then by *correctness* of $\mathcal{E}$ we have that $\mathcal{E}.\mathsf{Dec}(\mathsf{dsk}, \boldsymbol{c}_i^{(1)})$ is equal to $\boldsymbol{m}_i||u_i$, thus $\mathcal{H}(u_i) = \boldsymbol{c}_i^{(2)}$ and so $\boldsymbol{e}[i] = \boldsymbol{s}[i] = 1$.
2. If $i \in \mathcal{S}_0$, $\boldsymbol{c}_i$ is generated as Eq. (1) from oblivious sampling. Below we expound that if $\boldsymbol{e}[i]$ is assigned as 0 with non-negligible probability ϵ, then we can break the one-wayness of $\mathcal{H}$ with the same probability ϵ. A PPT adversary $\mathcal{A}$ first generates $(\mathsf{pk}_{\mathcal{A}}, \mathsf{sk}_{\mathcal{A}}) \leftarrow \mathcal{E}.\mathsf{KGen}(1^\lambda)$, then produces a random cipher $\boldsymbol{c}_{\mathcal{A}} \leftarrow \mathcal{E}.\mathsf{OEnc}(\mathsf{pk}_{\mathcal{A}}; r)$. Next, $\mathcal{A}$ requests a challenge for the one-wayness game and receives $\mathcal{H}(u)$ for random $u \in \mathcal{U}$, and decrypts $\boldsymbol{c}_{\mathcal{A}}$ as $\overline{\boldsymbol{m}}_{\mathcal{A}}$ using $\mathsf{sk}_{\mathcal{A}}$. If $\overline{\boldsymbol{m}}_{\mathcal{A}} = \perp$, $\mathcal{A}$ also outputs $\perp$ and aborts. Else, $\mathcal{A}$ parses $\overline{\boldsymbol{m}}_{\mathcal{A}}$ as $\boldsymbol{m}'||u'$ and outputs u'. Note that $(\boldsymbol{c}_{\mathcal{A}}, \mathcal{H}(u))$ is generated in the same way as Eq. (1), meaning the success of $\mathcal{A}$ in the one-wayness game of $\mathcal{H}$ (i.e., $\mathcal{H}(u') = \mathcal{H}(u)$) is equivalent to assigning $\boldsymbol{e}[i]$ to 1 in this sub-case.

After the above analysis, we have $\boldsymbol{e} = \boldsymbol{s}$ holds with overwhelming probability, so $t_{\boldsymbol{e}} = t = \mathsf{L}(\boldsymbol{s},\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}))$ and $\mathcal{DE}.\mathsf{Dec}$ always outputs $\boldsymbol{m}_{t_{\boldsymbol{e}}} = \boldsymbol{m}$. □

3.3 Security Analysis

Below we prove $\mathcal{DE}$ satisfies *IND-CPA* and $\frac{1}{\sqrt{n}}$-*deniability*.

Theorem 3. *Suppose that $\mathcal{E}$ is IND-CPA, then $\mathcal{DE}$ is IND-CPA.*

Proof. We prove *CPA security* by contradiction. Suppose that $\mathcal{A}$ succeeds in $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{CPA-b}}$ of $\mathcal{DE}$ with probability $\frac{1}{2} + \epsilon$ for non-negligible ϵ, then we can build a PPT algorithm $\mathcal{B}$ that breaks *CPA security* of $\mathcal{E}$ with also advantage ϵ. Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{E}.\mathsf{KGen}(1^\lambda)$, given pk, $\mathcal{B}$ plays with $\mathcal{A}$ as follows:

- **Setup.** $\mathcal{B}$ samples $\mathsf{k}_F \leftarrow \mathcal{F}.\mathsf{Key}(1^\lambda)$, then sends $\mathsf{dpk} = (\mathsf{pk}, \mathsf{k}_F)$ to $\mathcal{A}$.
- **Challenge.** $\mathcal{A}$ picks two different messages $\boldsymbol{m}_0, \boldsymbol{m}_1 \leftarrow \mathcal{M}$ and submits them to $\mathcal{B}$. Then $\mathcal{B}$ computes $h'_0 = \mathcal{H}(\boldsymbol{m}'_0)$ and $h'_1 = \mathcal{H}(\boldsymbol{m}'_1)$. Next, $\mathcal{B}$ sends $(\boldsymbol{m}_0||u, \boldsymbol{m}_1||u)$ to the challenger, where $u \leftarrow \mathcal{U}$. In this way, the challenger flips a random coin $b \in \{0,1\}$, picks the internal randomness $r \leftarrow \mathcal{R}$ and outputs a challenging ciphertext $\boldsymbol{c} \leftarrow \mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}_b||u_b; r)$. Finally, $\mathcal{B}$ performs as $\mathcal{DE}.\mathsf{Enc}$ to produce the trick ciphertexts for $\mathcal{A}$ as follows:
  1. Pick $\boldsymbol{s}, \boldsymbol{v} \leftarrow \{0,1\}^n$. Abort if $\boldsymbol{s} = 0^n$; Else, set $t = \mathsf{L}(\boldsymbol{s},\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}))$.
  2. Set $\boldsymbol{c}_t := (\boldsymbol{c}, \mathcal{H}(u))$, and for $i \in [n] \setminus t$, do the following:
     ① If $i \in \mathcal{S}_0$, pick $r_i, u_i \leftarrow \mathcal{R} \times \mathcal{U}$ and obtain $\boldsymbol{c}_i \leftarrow (\mathcal{E}.\mathsf{OEnc}.(\mathsf{pk}; r_i), \mathcal{H}(u_i))$.
     ② Else, sample $\boldsymbol{m}_i \leftarrow \mathcal{M}$ and $r_i, u_i \leftarrow \mathcal{R} \times \mathcal{U}$, then generate $\boldsymbol{c}_i \leftarrow (\mathcal{E}.\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}_i||u_i; r_i), \mathcal{H}(u_i))$.
  3. Return $\mathsf{dct} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{v})$ to $\mathcal{A}$.
- **Guess.** $\mathcal{A}$ outputs a guess bit b', $\mathcal{B}$ also outputs b' as the guess of b.

From the above construction, we know that the only difference between the distributions of $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{CPA-0}}$ and $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{CPA-1}}$ is the target ciphertext $\boldsymbol{c}$. Thus, the fact that $\mathcal{A}$ wins with probability $\frac{1}{2} + \epsilon$ implies that $\mathcal{B}$'s advantage of breaking *CPA security* of $\mathcal{E}$ is also ϵ, which concludes the proof. □

Theorem 4. *$\mathcal{DE}$ is $\frac{1}{\sqrt{n}}$-deniable.*

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be PPT algorithms, playing the role of adversary and challenger in $\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De-b}}$, respectively. For a fake claim under coercion, consider the following hybrid games, where R_i is the output of the adversary in game i.

Game 0. This is the honest encryption case, the distribution from $\mathcal{A}$'s view is

$$D_0 = (\mathsf{dpk}, \boldsymbol{m}^*, \mathsf{Rand}, \mathsf{ct}_0),$$

where $\mathsf{ct}_0 \leftarrow \mathcal{DE}.\mathsf{Enc}(\mathsf{dpk}, \boldsymbol{m}^*; \mathsf{Rand}, \boldsymbol{v})$, Rand and $\boldsymbol{v}$ are sampled as follows:

1. Pick $\boldsymbol{s}, \boldsymbol{v} \leftarrow \{0,1\}^n$. Abort if $\boldsymbol{s} = 0^n$; Else, set $t = \mathsf{L}(\boldsymbol{s},\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}))$.
2. For $i \in [n]$: sample $r_i \leftarrow \mathcal{R}, u_i \leftarrow \mathcal{U}$; further if $i \in \mathcal{S}_1$, sample $\boldsymbol{m}_i \leftarrow \mathcal{M}$, except that take $\boldsymbol{m}_t = \boldsymbol{m}$ for $i = t$.
3. Return $\mathsf{Rand} := (\boldsymbol{s}, \{\boldsymbol{m}_i\}_{i \in \mathcal{S}_1 \setminus \{t\}}, \{r_i, u_i\}_{i \in [n]})$ and $\boldsymbol{v}$.

Game 1. This game turns to generate the randomness Rand' and $\boldsymbol{v}$ as follows:

1. Select $\boldsymbol{s}, \boldsymbol{v} \leftarrow \{0,1\}^n$. Abort if $\boldsymbol{s} = 0^n$; Else, set $t = \mathsf{L}(\boldsymbol{s},\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}))$.
2. Flip $\boldsymbol{s}$ into $\boldsymbol{s}' = (\ldots s_{t-1}\ 0\ s_{t+1}\ \ldots)$, if $\boldsymbol{s}' = 0^n$, which occurs with negligible probability $\frac{n}{2^n}$, abort; else, set $t' = \mathsf{L}(\boldsymbol{s}',\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}'))$.
3. For $i \in [n]$: sample $r_i^{'} \leftarrow \mathcal{R}, u_i^{'} \leftarrow \mathcal{U}$, further if $i \in \mathcal{S}'_1 \setminus \{t'\}$, sample $\boldsymbol{m}'_i \leftarrow \mathcal{M}$.
4. Return $\mathsf{Rand}' := (\boldsymbol{s}', \{\boldsymbol{m}'_i\}_{i \in \mathcal{S}'_1 \setminus \{t'\}}, \{r_i^{'}, u_i^{'}\}_{i \in [n]})$ and $\boldsymbol{v}$.

In this way, the output distribution from $\mathcal{A}$'s view is

$$D_1 = (\mathsf{dpk}, \boldsymbol{m}^*, \mathsf{Rand}', \mathsf{ct}_1),$$

where $\mathsf{ct}_1 \leftarrow \mathcal{DE}.\mathsf{Enc}(\mathsf{dpk}, \boldsymbol{m}^*; \mathsf{Rand}', \boldsymbol{v})$. Note D_0 and D_1 only differ in the random seed $\boldsymbol{s}$ and $\boldsymbol{s}'$. Further, the distance between the distribution of $\boldsymbol{s}'$ and

F in Thm. 1 is at most $\frac{n}{2^n}$, which is exactly the maximum difference between selecting the flipping index t from random and as $\mathsf{L}(\boldsymbol{s}',\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}'))$. In this sense, we conclude that $\mathsf{SD}(\boldsymbol{s},\boldsymbol{s}')$ is $\Theta(\frac{1}{\sqrt{n}})$. Hence, it holds that $\Pr[R_1 = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-0} = 1] \leq \Theta(\frac{1}{\sqrt{n}})$.

Game 2. This is the faking case, the distribution from $\mathcal{A}$'s view is

$$D_2 = (\mathsf{dpk}, \boldsymbol{m}^*, \mathsf{Rand}^*, \mathsf{ct}_2),$$

where $\mathsf{ct}_2 \leftarrow \mathcal{DE}.\mathsf{Enc}(\mathsf{dpk}, \boldsymbol{m}; \mathsf{Rand}, \boldsymbol{v})$, the real randomness Rand and $\boldsymbol{v}$ are sampled in the same way as that in Game 0, while the fake randomness Rand^* is sampled as $\mathcal{DE}.\mathsf{Fake}$ operates:

1. Set $\boldsymbol{s}^* = (\ldots \boldsymbol{s}[t-1]\ 0\ \boldsymbol{s}[t+1]\ldots)$ and $t^* = \mathsf{L}(\boldsymbol{s}^*,\ \mathsf{dec}(\boldsymbol{v}) \bmod \mathsf{w}(\boldsymbol{s}^*))$.
2. For $i \in [n]$: if $i = t$, generate $r_i^* := \mathcal{E}.\mathsf{IEnc}(\mathsf{pk}, \boldsymbol{m}, r_i)$ and set $u_i = u_i$; Else, set $r_i^* = r_i$ and $u_i^* = u_i$, additionally set $\boldsymbol{m}_i^* = \boldsymbol{m}_i$ if $i \in \mathcal{S}_1 \setminus \{t^*\}$.
3. Return $\mathsf{Rand}^* = (\boldsymbol{s}^*, \{\boldsymbol{m}_i^*\}_{i\in\mathcal{S}_1\setminus\{t,t^*\}}, \{r_i^*, u_i^*\}_{i\in[n]})$.

After the above steps, ct_2 can also be explained as $\mathcal{DE}.\mathsf{Enc}(\mathsf{dpk}, \boldsymbol{m}^*; \mathsf{Rand}^*)$. Therefore, the only difference between D_1 and D_2 is the fake randomness Rand^* and Rand'. To evaluate this distance, consider their components:

a). $\boldsymbol{s}$ is uniformly random over $\{0,1\}^n$, $\boldsymbol{s}^*$ is indeed sampled from one-bit flipping frame F. Thus, $\mathsf{SD}(\boldsymbol{s},\boldsymbol{s}^*) \leq \frac{n}{2^n} = \mathsf{negl}(\lambda)$, as the above game scales.

b). All the masking messages and the relative randomness are uniformly random over $\mathcal{M}\times\mathcal{R}\times\mathcal{U}$, for $i \in [n] \setminus t$.

c). r_t^* from $\mathcal{E}.\mathsf{IEnc}$ is computationally indistinguishable from $r_t' \in \mathcal{R}$, both u_t^* and u_t' are uniformly random over $\mathcal{U}$.

The above shows Rand^* and Rand' are computationally indistinguishable from each other. Hence, it holds that $\Pr[R_1 = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-1} = 1] = \mathsf{negl}(\lambda)$.

Taking in all the cases, we have $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-1} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathsf{De}-0} = 1] \leq \Theta(\frac{1}{\sqrt{n}})$, so the theorem holds. □

4 SCA on Deniable Encryption

As noted in Sec. 1, none of the existing DPKE considers the issue of SCA. In this section, we make an initial attempt towards leakage-resilient DPKE. We begin with formalizing the SCA-equipped coercion model for timing and simple power attacks [26,27,30,35], then show how such SCA could break deniability of known schemes and ours, along with giving some heuristic countermeasures.

4.1 SCA-equipped Coercion Model

In the original attack model [7], the coercer *Eve* first intercepts a dispatched package (ciphertext) from the sender *Alice*, then obtains the claimed plaintext and randomness from *Alice*. In this sense, deniability (Def. 2) asks that *Eve* has no extra advantage in distinguishing between the honest and fake opening. Now, *Eve* can resort to SCA when performing attacks. In particular, *Eve* can additionally collect the SC information (time or power consumption) about the operations of the original encryption and that under the claimed data. Below we formalize this enhanced coercion model for *Eve*.

Definition 4 (SCA-Coercion Model). *For any deniable public key encryption system, a coercer can perform the following attack steps to a system user:*

1. *Passively capture the transmitted ciphertext* ct *and* SC *information* $\mathcal{T}$ *(e.g., time or power consumption) of the encryption execution that produces* ct*;*
2. *Actively demand the internal message* $\boldsymbol{m}$ *and randomness* r *relative to* ct *and collect new* SC *information* $\overline{\mathcal{T}}$ *of the encryption execution on feeding* $(\boldsymbol{m}, r)$*.*

Remark 2. We assume that there are no external operations, e.g., ones profiled in more advanced trace [32] or collision attacks [25], are to be executed in running deniable encryption. Besides, to avoid systematic error, a coercer may demand the posterior SC information $\mathsf{poly}(\lambda)$ times.

4.2 Break Deniability of Known Schemes

Below we depict how deniable schemes can be breached under the above enhanced hostile model. The core point is that we observe the internal instruction lines take on some constant difference between the original call of encryption and that of fake opening, which will result in the perceptible gap between $\mathcal{T}$ and $\overline{\mathcal{T}}$. In this sense, the coercer can use such flavor of distinguishability to tell if a user is lying, details of these attacks to the known schemes are as follows:

- *Translucent-set-based.* Note that the instantiations (e.g., trapdoor permutation [7], simulatable encryption, or lattice-based methods [33]) of translucent set $\mathcal{S}$, are much more complicated than the uniform random set $\mathcal{R}$ which can be built-in. Based on this fact, sampling from $\mathcal{S}$ always takes more operation than sampling from $\mathcal{R}$, indicating more time or power is consumed. Then the coercer can tell that the sender is lying if $\mathcal{T}$ is statistically higher than $\overline{\mathcal{T}}$. More specifically, consider the pioneering work [7] (Fig. 1) for bit encryption.
  - The sender encodes the bit into the parity of the number i of $\mathcal{S}$-elements. To fake, the sender just claims to have chosen $i' = i - 1$. Then the count of $\mathcal{S}$-elements during the rerun of encryption always decreases by 1. Hence, the coercer can first get the prior value $\mathcal{T}$ and many posterior values $\{\overline{\mathcal{T}}\}$, and decides that the claimed randomness is fake if the percentage of $\{\overline{\mathcal{T}}\}$ which is lower than $\mathcal{T}$ is significant, e.g., $> 80\%$.

Notations: $\mathcal{R} = \{0,1\}^t$; translucent set $\mathcal{S} \subset \{0,1\}^t$; $\mathcal{S}$'s trapdoor $d_{\mathcal{S}}$

$\mathsf{Enc}(\mathcal{S}, m)$: For $m = 0$ (resp., $m = 1$), pick a random even (resp., odd) number $i \in [n]$, sample $s_1, \ldots, s_i \leftarrow \mathcal{S}$ and $r_{i+1}, \ldots, r_\lambda \leftarrow \mathcal{R}$, output $\mathsf{ct} := (s_1, \ldots, s_i, r_{i+1}, \ldots, r_\lambda)$.

$\mathsf{Dec}(\mathcal{S}, d_{\mathcal{S}}, \mathsf{ct})$: Output the parity of the number of $\mathcal{S}$-element in ct via $d_{\mathcal{S}}$.

$\mathsf{Fake}(\mathcal{S}, m, i, \overline{m})$: If $i = 0$, cheating fail, otherwise output $i - 1$.

Fig. 1. Sketch of the scheme in [7].

- *iO-based* [34]. From Fig. 2, we know that the honest encryption executes step 3 of $\mathsf{Encrypt}$ which is a call of the underlying PKE, while the faking randomness

leads to step 2 which are just two evaluations of two PRFs. This fact implies that the prior time or power consumption $\mathcal{T}$ is always higher than the posterior one $\overline{\mathcal{T}}$ even under the obfuscated setting (recall that $i\mathcal{O}$ only ensures the obfuscated programs for circuits of the same size and functionality are indistinguishable.). Thus, the coercer could apply the same strategy as above, i.e., demand τ posterior values $\{\overline{\mathcal{T}}\}$ with respect to the claimed data $(\overline{m}, r')$ and identify the lie if 80% of $\{\overline{\mathcal{T}}\}$ is lower than $\mathcal{T}$.

<table>
<tr><td colspan="2">Notations: PKE scheme $\mathcal{E}' = \{\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}\}$; Puncturable PRFs F_1, F_2, F_3</td></tr>
<tr><td colspan="2">$\mathsf{DKGen}(1^\lambda)$: Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, $P_{enc} = i\mathcal{O}(\mathsf{Encrypt})$, $P_{exp} = i\mathcal{O}(\mathsf{Explain})$, output $\mathsf{dpk} = (P_{enc}, P_{exp})$ and $\mathsf{dsk} = \mathsf{sk}$.
$\mathsf{DEnc}(\mathsf{dpk}, m)$: Sample $r \leftarrow \{0,1\}^{\mathsf{poly}(\lambda)}$, output $\mathsf{dct} \leftarrow P_{enc}(m, r)$.
$\mathsf{DDec}(\mathsf{dsk}, \mathsf{dct})$: Output $m := \mathsf{Dec}(\mathsf{dsk}, \mathsf{dct})$.
$\mathsf{Fake}(\mathsf{dpk}, m, r, \overline{m})$: Output $r' \leftarrow P_{exp}(\mathsf{dct}, m, r)$.</td></tr>
<tr><td>Algorithm $\mathsf{Encrypt}(m, r)$

1. Parse $r = r_1 || r_2$.
2. If $F_3(r_1) \oplus r_2 = m||c||r'$ and $r_1 = F_2(m||c||r')$, output c and stop.
3. Output $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; F_1(m||r))$.</td><td>Algorithm $\mathsf{Explain}$(c,m,r)

– Set $\alpha = F_2(m||c||PRG(r))$, $\beta = F_3(\alpha) \oplus m||c||PRG(r)$, output $\alpha||\beta$.</td></tr>
</table>

Fig. 2. Sketch of the schemes in [34].

- *FHE-based.* The FHE-based design for bits [2] modifies the line of [7] via building a biased decryption to "0" on random input, while it still encodes the bit message into the parity of the count i of "1"-encryptions in the final ciphertext. To fake, the sender reveals $i' = i - 1$ by randomly interpreting one of "1"-encryptions into "0"-encryption. Further, compared with a "1"-encryption that needs several homomorphic evaluations, a "0"-encryption is done by a random sampling, thus the time consumption of dishonest opening must be less than that of the honest case, implying $\mathcal{T}$ is always higher than $\overline{\mathcal{T}}$.
- *Our schemes.* The issue of $\mathcal{DE}$ is akin to that of [7]. Encryption at "1" invokes a more time (power)-consuming call of the encryption of the underlying *CS*-PKE, than the single oblivious sampling for encryption at "0". To fake, the sender reveals $\boldsymbol{s}^*$, whose count of "1" always decreases by 1 than that of the real $\boldsymbol{s}$, leading to $\overline{\mathcal{T}}$ being lower than $\mathcal{T}$ with overwhelming probability.

Experimental results. To evidence that the above attacks are workable, we instantiate the scheme of [7] and ours with ElGamal over $\mathbb{Z}_q$ in Python on the Intel Kaby Lake i7-7700T processor, where we use CPU instruction "rdtsc" to compute the consumption of clock cycles (3.6 GHz) for an encryption execution. More detailedly, for a TS, $\mathcal{S}$-element is a triple $(m, c_1, c_2) \in \mathbb{Z}_q^3$ where (c_1, c_2) is the ElGamal encryption of m, and $\mathcal{R}$-element a random triple over $\mathbb{Z}_q^3$; Our scheme is derived from the ciphertext-simulatability of ElGamal, i.e., OEnc out-

puts $(c_1, c_2) \leftarrow \mathbb{Z}_q^2$ and IEnc trivially simulates sampling over $\mathbb{Z}_q^2$ (see [16]).

For parameters, we set $\lambda = 128$, $\log q = 2048$, $\ell_h = 1024, \ell_t = 512, \ell_m = 1024$, OWF $\mathcal{H}$ as SHA3-512. Then for detecting probability $n = 2^{2k}, k \in [5, 15]$, we take 10^3 times of random encryption execution for both schemes, and term one execution as a success for a coercer if $\mathcal{T} - \overline{\mathcal{T}} \geq \mu + \delta$ ms, where $\mu = 0.0506$ is the expected difference of time consumption between one execution of ElGamal encryption and random sampling from $\mathbb{Z}_q^2$, and $\delta = 0.0040$ is the system error.

$\log n$	10	12	14	16	18	20	22	24	26	28	30
*succ_prob_*I	87.2	91.0	94.6	91.7	96.6	94.3	92.7	82.5	94.6	91.8	94.2
*cont_prob_*I	3.7	3.4	8.3	5.6	9.2	3.5	1.1	1.3	7.4	4.1	7.3
*succ_prob_*II	92.4	99.5	88.1	80.8	92.3	90.6	86.7	90.2	83.0	91.4	95.1
*cont_prob_*II	8.3	1.5	6.7	12.6	2.1	11.3	6.5	3.8	9.0	10.9	1.2

Table 2. Success probability (%) of SCA-equipped coercion attacks for different n. **I**: scheme in [7]. **II**: our scheme.

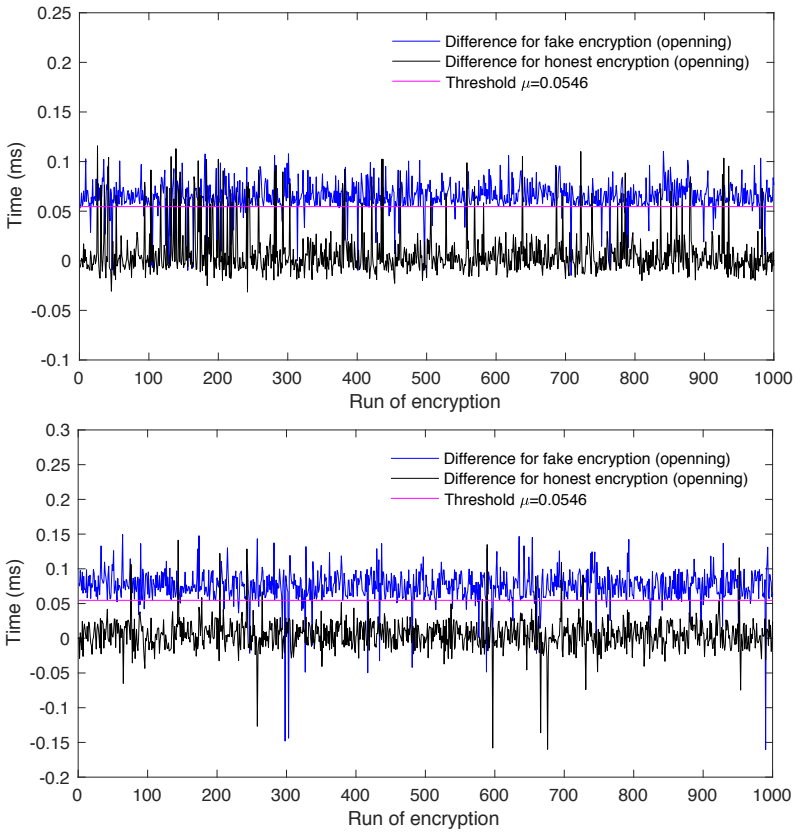

Fig. 3. Distributions of difference in time consumption for $n = 2^{30}$. **Top**: scheme in [7]. **Bottom**: our scheme.

Table 2 lists the running results of a simulated coercer who is equipped with SCA, rows "cont_prob I/II" represent the success probability for the control experiment where each $\overline{\mathcal{T}}$ is recorded feeding the real randomness and plaintext used for encryption (i.e., the sender is honest). In particular, for $n = 2^{30}$, Fig. 3 shows the distributions of the difference in time consumption between the original call and the honest/fake opening (black/blue colored) of 10^3 encryption executions, where the trace for fake is apparently under the threshold $\mu = 0.0546$ ms, while the trace for honesty mainly fluctuates around the zero point[1]. Further, from Table 2, we learn that: 1). the success probability for a fake opening is significant ($> 80\%$); 2) the success probability for an honest opening (the control group) is inappreciable ($< 15\%$). Then we can conclude that the above-described attacks under the enhanced model (Def. 4) are practically effective, signifying the damage of deniability from SCA.

4.3 Feasible Therapies

One can take some random and functionless instructions on the hardware layer to perturb SC information. However, as noted in [26], such system noise can be compensated by collecting more records. A more substantial way is to add some redundancy operations into the original encryption algorithm, making honesty and faking execute the very same instruction stream, so to eliminate the difference in the context of SC knowledge. Below we give the concrete fixing:

- *Translucent-set-based.* To encrypt a bit m, first sample λ elements $\{s_i\}$ from $\mathcal{S}$ and λ elements $\{r_i\}$ from $\mathcal{R}$, then pick a random even (resp., odd) number $i \in [n]$ for $m = 0$ (resp., $m = 1$) and output $\mathsf{ct} := (s_1, \ldots, s_i, r_{i+1}, \ldots, r_\lambda)$.
- *iO-based.* Let step 2 of Encrypt in Fig. 2 additionally conduct a plain encryption $\mathcal{E}.\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}; \mathcal{F}_2.\mathsf{Eval}(\mathsf{k}_2, \boldsymbol{m}||r))$, which is exactly what step 3 executes.
- *FHE-based*: For the scheme [2], always produce n encryptions of bit 1 and sample n random elements from the ciphertext space of the underlying FHE.
- *Our scheme.* Both encryptions at "1" and "0" now conduct a plain encryption and an oblivious sampling. Namely, in algorithm Enc, step 3. additionally performs $\boldsymbol{m}_i \leftarrow \mathcal{M}, \boldsymbol{c}_i^{(3)} \leftarrow \mathcal{E}.\mathsf{Enc}(\mathsf{pk}, \boldsymbol{m}_i||u_i; r_i)$, and step 3. extraly runs $\boldsymbol{c}_i^{(3)} \leftarrow \mathcal{E}.\mathsf{OEnc}.(\mathsf{pk}; r_i)$. The sender will not transmit the auxiliary ciphertexts $\{\boldsymbol{c}_i^{(3)}\}$ and just keep the masks $(\{\boldsymbol{m}_i\}_{i\in\mathcal{S}_0}, \{\boldsymbol{c}_i^{(3)}\}_{i\in[n]})$ in her internal state.

To show the feasibility of these measures, we carry the above experiments to the upgraded variants of the scheme [7] and ours, and profile the results in Table 3 and Fig. 4 below, where we can learn that now the success probability for a fake opening is also reduced to be invisible ($< 15\%$), and the time consumptions of honesty and fake are almost the same as that of the original encryption (both traces fluctuates around zero). These facts testify that the redundancy operations really conceal the difference in SC information between the honest encryption and fake opening, so that *deniability* maintains.

[1] Due to the page limits, we omit the graphics for other values of n that show the similar grades as that of $n = 2^{30}$.

$\log n$	5	6	7	8	9	10	11	12	13	14	15
*succ_prob_*I	1.1	0.8	3.7	4.8	8.5	1.2	5.3	4.9	10.8	2.9	0.8
*cont_prob_*I	2.4	1.6	4.3	3.3	7.2	2.5	4.6	8.3	13.4	4.1	0.6
*succ_prob_*II	3.7	3.9	6.2	8.1	4.6	1.4	8.8	0.7	5.9	7.0	2.7
*cont_prob_*II	5.1	1.7	3.0	11.9	7.3	5.6	7.6	3.9	8.2	3.9	4.4

Table 3. Success probability (%) of SCA-equipped coercion attacks for different n. **I**: upgraded variant of scheme in [7]. **II**: upgraded variant of our scheme.

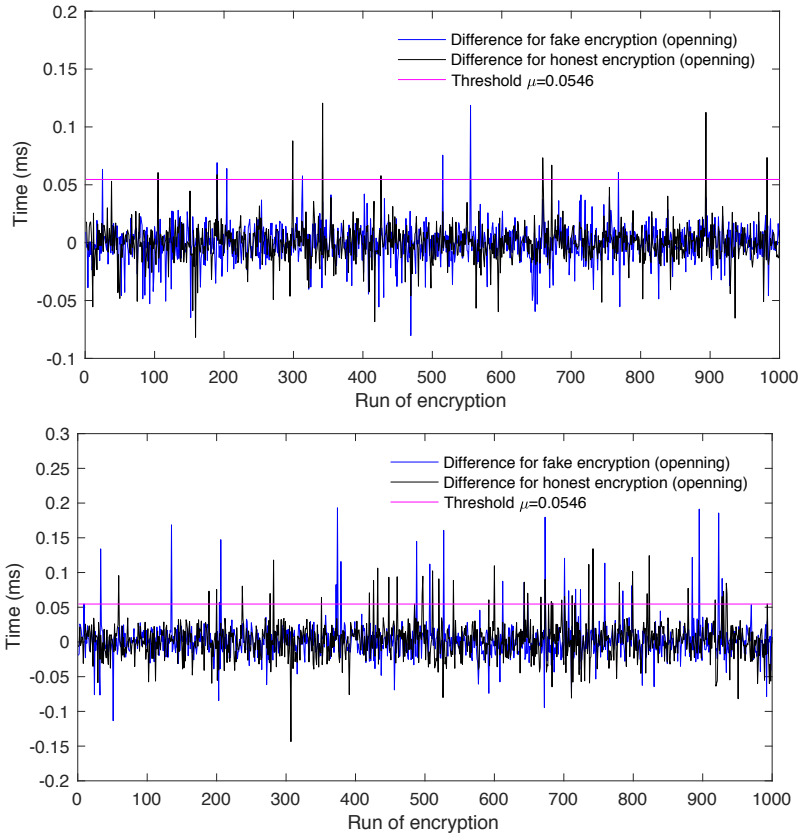

Fig. 4. Distributions of difference in time consumption for $n = 2^{30}$. **Top**: upgraded variant of scheme in [7]. **Bottom**: upgraded variant of our scheme.

Acknowledgments. This work is supported by the National Key R&D Program of China under Grant (2022YFB2701500) and Guangdong Major Project of Basic and Applied Basic Research (2019B030302008) and the National Natural Science Foundation of China (No. 62272491, and No. 61972429).

References

1. Aciiçmez, O., Brumley, B.B., Grabher, P.: New results on instruction cache attacks. In: Mangard, S., Standaert, F. (eds.) CHES 2010. LNCS, vol. 6225, pp. 110–124. Springer (2010)
2. Agrawal, S., Goldwasser, S., Mossel, S.: Deniable fully homomorphic encryption from learning with errors. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12826, pp. 641–670. Springer (2021)
3. Apon, D., Fan, X., Liu, F.: Deniable attribute based encryption for branching programs from LWE. In: Hirt, M., Smith, A.D. (eds.) TCC 2016. LNCS, vol. 9986, pp. 299–329 (2016)
4. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 280–300. Springer (2013)
5. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory **6**(3), 13:1–13:36 (2014)
6. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-or-nothing transforms. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 453–469. Springer (2000)
7. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Jr., B.S.K. (ed.) CRYPTO 97. vol. 1294, pp. 90–104. Springer (1997)
8. Canetti, R., Park, S., Poburinnaya, O.: Fully deniable interactive encryption. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12170, pp. 807–835. Springer (2020)
9. Cao, Y., Zhang, F., Gao, C., Chen, X.: New practical public-key deniable encryption. In: Meng, W., Gollmann, D., Jensen, C.D., Zhou, J. (eds.) ICICS 2020. LNCS, vol. 12282, pp. 147–163. Springer (2020)
10. Caro, A.D., Iovino, V., O'Neill, A.: Deniable functional encryption. In: Cheng, C., Chung, K., Persiano, G., Yang, B. (eds.) PKC 2016. LNCS, vol. 9614, pp. 196–222. Springer (2016)
11. Caro, A.D., Iovino, V., O'Neill, A.: Receiver- and sender-deniable functional encryption. IET Inf. Secur. **12**(3), 207–216 (2018)
12. Chi, P., Lei, C.: Audit-free cloud storage via deniable attribute-based encryption. IEEE Trans. Cloud Comput. **6**(2), 414–427 (2018)
13. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: A homomorphic LWE based e-voting scheme. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 245–265. Springer (2016)
14. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: Improved non-committing encryption with applications to adaptively secure protocols. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 287–302. Springer (2009)
15. Coladangelo, A., Goldwasser, S., Vazirani, U.V.: Deniable encryption in a quantum world. In: Leonardi, S., Gupta, A. (eds.) STOC 2022. pp. 1378–1391. ACM (2022)
16. Damgård, I., Nielsen, J.B.: Improved non-committing encryption schemes based on a general complexity assumption. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 432–450. Springer (2000)
17. Dent, A.W.: The cramer-shoup encryption scheme is plaintext aware in the standard model. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 289–307. Springer (2006)

18. Dodis, Y., Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer (2010)
19. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008. pp. 293–302. IEEE Computer Society (2008)
20. Gao, C., Xie, D., Wei, B.: Deniable encryptions secure against adaptive chosen ciphertext attack. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 46–62. Springer (2012)
21. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS 2013. pp. 40–49. IEEE Computer Society (2013)
22. Garg, S., Polychroniadou, A.: Two-round adaptively secure MPC from indistinguishability obfuscation. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 614–637. Springer (2015)
23. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) STOC 2009. pp. 169–178. ACM (2009)
24. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. In: Khuller, S., Williams, V.V. (eds.) STOC 2021. pp. 60–73. ACM (2021)
25. Kaminsky, D., Patterson, M.L., Sassaman, L.: PKI layer cake: New collision attacks against the global X.509 infrastructure. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 289–303. Springer (2010)
26. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer (1996)
27. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer (1999)
28. Li, H., Zhang, F., Fan, C.: Deniable searchable symmetric encryption. Inf. Sci. **402**, 233–243 (2017)
29. Matsuda, T., Hanaoka, G.: Trading plaintext-awareness for simulatability to achieve chosen ciphertext security. In: Cheng, C., Chung, K., Persiano, G., Yang, B. (eds.) PKC 2016. LNCS, vol. 9614, pp. 3–34. Springer (2016)
30. Messerges, T.S.: Using second-order power analysis to attack DPA resistant software. In: Koç, Ç.K., Paar, C. (eds.) CHES 2000. LNCS, vol. 1965, pp. 238–251. Springer (2000)
31. Micali, S., Reyzin, L.: Physically observable cryptography (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer (2004)
32. Michalevsky, Y., Schulman, A., Veerapandian, G.A., Boneh, D., Nakibly, G.: Powerspy: Location tracking using mobile device power analysis. In: Jung, J., Holz, T. (eds.) USENIX Security Symposium 2015. pp. 785–800. USENIX Association (2015)
33. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 525–542. Springer (2011)
34. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) STOC 2014. pp. 475–484. ACM (2014)
35. Silverman, J.H., Whyte, W.: Timing attacks on ntruencrypt via variation in the number of hash calls. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 208–224. Springer (2007)

A Towards Optimal Flipping Sampling

One natural question raised from the above design of DPKE is: are there any other ways of flipping bits of $\boldsymbol{s}$ that result in a closer distance from U, leveraging which we can devise a DPKE with better deniability? Below we give the negative answer by showing that flipping one bit is actually the optimal way, by proving that it is superior to any t-bit flipping ($t > 1$) or uniformly random flipping. For simplicity, hereafter we assume $\binom{n}{k} = 0$ for $k > n$.

Theorem 5. *For $t \in [1, n]$, let F_t be the flipping case where it first samples $\boldsymbol{s}$ from $\mathcal{S}$, if the count of 1 in $\boldsymbol{s}$ is less than t, outputs $\perp$; else randomly flips t bits in $\boldsymbol{s}$ from 1 to 0. It holds $\mathsf{SD}(U, F_t) > \mathsf{SD}(U, F)$ for $t \geq 2$.*

Proof. Observe that $\boldsymbol{s}$ must be obtained by flipping t bit 1 of some string $\boldsymbol{s}'$ from $\mathcal{S}$ whose count of bit 1 is $k + t$. Thus, there are $\binom{n-k}{t}$ possible $\boldsymbol{s}'$ when fixing $\boldsymbol{s}$. Further, the probability of exactly flipping the corresponding 1 of $\boldsymbol{s}'$ is $1/\binom{k+t}{t}$. Then $\forall \boldsymbol{s} \in \mathcal{S}, F(\boldsymbol{s}) = \frac{1}{2^n} \cdot \binom{n}{k}\binom{n-k}{t}/\binom{k+t}{t}$, and the distance between R and F_t is

$$\begin{aligned} \mathsf{SD}(R, F_t) &= \frac{1}{2} \cdot \sum_{k=0}^{n} \left| \frac{1}{2^n}\binom{n}{k}\left(1 - \frac{\binom{n-k}{t}}{\binom{k+t}{t}}\right)\right| + \frac{1}{2} \cdot F_t(\perp) \\ &= \frac{1}{2^{n+1}} \cdot \left(\sum_{k=0}^{n} \left| \binom{n}{k} - \binom{n}{k+t} \right| + \sum_{k=0}^{t-1} \binom{n}{k} \right). \end{aligned} \tag{2}$$

To prove $\mathsf{SD}(R, F_t) > \mathsf{SD}(R, F)$ for $t \geq 2$, it suffices to argue that $\mathsf{SD}(R, F_1)$ is the minimum value regarding $\mathsf{SD}(R, F_t)$ as a discrete function of t, for which we consider the following two cases:

– For $1 \leq t \leq m$, Eq. (2) can be simplified into $\frac{1}{2^n} \cdot \sum_{k=\lceil \frac{n-t}{2} \rceil}^{\lceil \frac{n+t}{2} \rceil - 1} \binom{n}{k}$, being monotonically increasing on t. So $t = 1$ is the minimum point in this interval.

– For $m + 1 \leq t \leq n$, Eq. (2) can be simplified into

$$\frac{1}{2^{n+1}} \cdot \left(\sum_{i=t}^{\lceil \frac{n+t}{2} \rceil - 1} \binom{n}{i} + \sum_{i=\lceil \frac{n-t}{2} \rceil}^{\lceil \frac{n+t}{2} \rceil - 1} \binom{n}{i} - \sum_{i=0}^{\lceil \frac{n-t}{2} \rceil - 1} \binom{n}{i} + \sum_{k=0}^{t-1} \binom{n}{k} \right).$$

To estimate the scale of the above equation, observe that

$$\left(\sum_{i=t}^{\lceil \frac{n+t}{2} \rceil - 1} \binom{n}{i} - \sum_{i=0}^{\lceil \frac{n-t}{2} \rceil - 1} \binom{n}{i} \right) \geq 0, \quad \left(\sum_{i=\lceil \frac{n-t}{2} \rceil}^{\lceil \frac{n+t}{2} \rceil - 1} \binom{n}{i} + \sum_{k=0}^{t-1} \binom{n}{k} \right) > 2 \cdot \binom{n}{m}.$$

Thus we can deduce that $\mathsf{SD}(R, F_t) > \mathsf{SD}(R, F)$ also holds in this interval. Based on the above analysis, it is clear that $\mathsf{SD}(R, F_t) > \mathsf{SD}(R, F)$ for $t \geq 2$. □

Theorem 6. *Let F' be the flipping case where it first samples $\boldsymbol{s}$ from $\mathcal{S} \setminus \{0^n\}$ and then randomly flips some bits of $\boldsymbol{s}$ (not all of 1) from 1 to 0, it holds* $\mathsf{SD}(U, F') > \mathsf{SD}(U, F)$.

Proof. Any $\boldsymbol{s}$ from F' must be obtained by flipping j bits 1 of some $\boldsymbol{s}'$ for $j \in [1, n-k]$, meaning the count of 1 of $\boldsymbol{s}'$ is $k+j$. So the generation of $\boldsymbol{s}$ can be divided into two steps: 1) choose the indexes of i bits 1 to fix $\boldsymbol{s}'$; 2) flip the target indexes of $\boldsymbol{s}'$. Hence, the total possible way of sampling $\boldsymbol{s}$ is $\frac{1}{2^n} \cdot \sum_{j=1}^{n-k} \binom{n-k}{j} \cdot 2^{-(k+j)}$. Then by traversing all the possible $\boldsymbol{s}$ ($\binom{n}{k}$ values), we have that

$$\begin{aligned} \mathsf{SD}(R, F') &= \frac{1}{2} \cdot \sum_{k=0}^{n} \left| \frac{1}{2^n} \binom{n}{k} \left(1 - \sum_{j=1}^{n-k} \binom{n-k}{j} \cdot 2^{-(k+j)} \right) \right| \\ &= \frac{1}{2^{n+1}} \cdot \sum_{k=0}^{n} \binom{n}{k} \left| 1 - 2^{-k} \cdot \sum_{j=1}^{n-k} \binom{n-k}{j} \cdot 2^{-j} \right|. \end{aligned} \tag{3}$$

To estimate the relative scale of Eq. (3), we first consider the item of the absolute value $\left| 1 - 2^{-k} \cdot \sum_{j=1}^{n-k} \binom{n-k}{j} \cdot 2^{-j} \right|$. Denote the sum of the involved sequence as $S_m = \sum_{j=0}^{m} \binom{m}{j} \cdot 2^{-j}$, a simple calculation shows that $S_{m+1} = \frac{3}{2} S_m$ (geometric progression), further arriving at the simplified expression $\left| 1 - \frac{3^{n-k}}{2^n} + \frac{1}{2^k} \right|$. For large n, e.g., $n > 2^5$, we obtain the following inequality:

$$\sum_{k=0}^{n} \binom{n}{k} \left| 1 - \frac{3^{n-k}}{2^n} + \frac{1}{2^k} \right| > \sum_{k=0}^{n} \binom{n}{k} \left| 1 - \frac{n-k}{k+1} \right|,$$

which implies that $\mathsf{SD}(R, F') > \mathsf{SD}(R, F)$. □