Standard Security Does Imply Security Against Selective Opening for Markov Distributions

Felix Heuer[1] Eike Kiltz[1] Krzysztof Pietrzak[2]

[1] Horst Görtz Institute for IT-Security, Ruhr-University Bochum, Germany
{felix.heuer,eike.kiltz}@rub.de
[2] Institute of Science and Technology, Austria
pietrzak@ist.ac.at

Abstract

About three decades ago it was realized that implementing private channels between parties which can be adaptively corrupted requires an encryption scheme that is secure against *selective opening attacks*. Whether standard (IND-CPA) security implies security against selective opening attacks has been a major open question since. The only known reduction from selective opening to IND-CPA security loses an exponential factor. A polynomial reduction is only known for the very special case where the distribution considered in the selective opening security experiment is a product distribution, i.e., the messages are samples independent from each other.

In this paper we give a reduction whose loss is quantified via the dependence graph (where message dependencies correspond to edges) of the underlying message distribution. In particular, for some concrete distributions including Markov distributions, our reduction is polynomial.

Keywords: public key encryption, selective opening security, Markov, IND-CPA, IND-SO-CPA

1 Introduction

Security under Selective Opening Attacks. Consider a scenario where many parties $1, \ldots, n$ send messages to one common receiver. To transmit a message $\mathbf{m}_i$, party i samples fresh randomness $\mathbf{r}_i$ and sends the ciphertext $\mathbf{c}_i = \mathsf{Enc}_{pk}(\mathbf{m}_i; \mathbf{r}_i)$ to the receiver. Assume an adversary $\mathcal{A}$ that does not only eavesdrop on the sent ciphertexts $(\mathbf{c}_1, \ldots, \mathbf{c}_n)$, but corrupts a set $\mathcal{I} \subseteq [n]$ of the sender's systems, thus learning the encrypted message $\mathbf{m}_i$ *and* the randomness $\mathbf{r}_i$ used to encrypt $\mathbf{m}_i$. The natural question to ask is whether the messages of uncorrupted parties remain confidential. Such attacks are referred to as *selective opening* (SO) *attacks* (under sender corruption).

Selective opening attacks naturally occur in multi-party computation where we assume secure channels between parties. Since a party might become corrupted we would need the encryption on the channels to be selective opening secure. In practice, the same argument applies to a server that establishes secure connections that shall remain secure if users are corrupted.

Difficulty of Proving Security under Selective Opening Attacks. The widely accepted standard notion for public-key encryption schemes is indistinguishability under chosen-plaintext attacks (IND-CPA security). At first sight one might consider a straight-forward hybrid argument to show that IND-CPA security already implies security against selective opening attacks since every party samples fresh randomness independently. Though, so far, nobody has been able to bring such a hybrid argument forward in general. Notice that revealing randomness $\mathbf{r}_i$ allows a selective opening adversary to verify that a corrupted ciphertext $\mathbf{c}_i$ is an encryption of $\mathbf{m}_i$. The adversaries possibility to corrupt parties introduces a difficulty in proving that standard (IND-CPA) security already implies selective opening security. It seems to be the case that the reduction has to know (i.e. guess) the complete set $\mathcal{I}$ of all corruptions going to be made by $\mathcal{A}$ in order to serve its security game *before* $\mathcal{A}$ actually announced the senders it would like to corrupt. Since $\mathcal{I}$ might be any subset of $\{1, \dots, n\}$, a direct approach would lead to an exponential loss in the reduction. A main technical obstacle is that the encrypted messages may depend on each other. If, for example, they are encrypted and sent sequentially, message $\mathbf{m}_i$ may depend on $\mathbf{m}_{i-1}$ and all previous messages. Thus, corrupting some parties might already leak some information on messages sent by not corrupted parties.

Until today, the only result in the standard model shows that IND-CPA implies selective opening security is by [8, 3] for the special case of a product distribution, i.e., all messages $\mathbf{m}_1, \dots, \mathbf{m}_n$ are sampled independently from each other. Intuitively, this holds since corrupting some ciphertext cannot reveal information on related messages if there are no related messages at all and the hybrid argument one might expect to work goes through. This leaves the following open question:

Does standard security imply selective opening security for any non-trivial message distribution?

1.1 Our Contributions

We present the first positive results in the standard model, showing that IND-CPA security implies IND-SO-CPA security for a class of non-trivial message distributions with few dependencies. Here IND-SO-CPA security refers to the indistinguishability-based definition of selective opening security usually referred to as *weak* IND-SO-CPA *security* (cf. [4]).

IND-SO-CPA requires that a passive adversary, that obtains ciphertexts $(\mathbf{c}_1, \dots, \mathbf{c}_n)$ and access to a ciphertext *opening* oracle, revealing the underlying message $\mathbf{m}_i$ of some ciphertext $\mathbf{c}_i$ and the randomness used to encrypt $\mathbf{m}_i$, cannot distinguish the originally encrypted messages from freshly resampled messages that are *as likely as the original messages* given the messages of opened ciphertexts.

We consider *graph-induced* distributions where dependencies among messages correspond to edges in a graph and show that IND-CPA implies IND-SO-CPA security for all graph-induced distributions that satisfy a certain *low connectivity* property.

In particular, our result holds for the class of Markov distributions, i.e. distributions on message vectors $(\mathbf{m}_1, \dots, \mathbf{m}_n)$ where all information relevant for the distribution of $\mathbf{m}_i$ is present in $\mathbf{m}_{i-1}$. We prove that any IND-CPA secure public-key encryption scheme is IND-SO-CPA secure if the messages are sampled from a Markov distribution.

For instance, our results cover distributions where message $\mathbf{m}_i$ contains all previous messages (e.g., email conversations), or distributions where messages are increasing, i.e., $\mathbf{m}_1 \leq \mathbf{m}_2 \leq \dots \leq \mathbf{m}_n$.

Details. Think of an n-message vector sampled from some distribution $\mathfrak{D}$ as a graph G on n vertices $\{1, \dots, n\}$ where we have an edge from message $\mathbf{m}_i$ to message $\mathbf{m}_j$ if the distribution of $\mathbf{m}_j$

depends on $\mathbf{m}_i$. Further, fix any subset $\mathcal{I} \subseteq \{1, \dots, n\}$ of opening queries made by some adversary.

The main observation is that removing $\mathcal{I}$ and all incident edges, G decomposes into connected components $C_1, \dots, C_{n'}$ that can be resampled independently, i.e. the distribution of messages on C_k solely depends on the messages in the neighbourhood of C_k and $\mathfrak{D}$.

To argue that there is no efficient adversary $\mathcal{A}_{\mathsf{SO}}$ that distinguishes sampled and resampled messages in the selective opening experiment, we proceed in a sequence of hybrid games, starting in a game where $\mathcal{A}_{\mathsf{SO}}$ obtains only sampled messages. In each hybrid step we use IND-CPA security to replace *sampled* messages on a connected component C_k with *resampled* messages without $\mathcal{A}_{\mathsf{SO}}$ noticing. To this end, the reduction from IND-CPA to the indistinguishability of two consecutive hybrids has to identify C_k to embed its own challenge before $\mathcal{A}_{\mathsf{SO}}$ makes any opening query.

We consider two approaches for guessing C_k. The first will take graphs into account that consist of polynomially many connected subgraphs, hence, the reduction can guess C_k right away. The second take studies graphs were every connected subgraph has a neighbourhood of constant size, allowing the reduction to guess C_k by guessing its neighbourhood. We show that the first approach ensures a reduction with polynomial loss for a strictly greater class of graphs than then second one.

Additionally, when the distribution is induced by an acyclic graph, we give a more sophisticated hybrid argument for the second approach, where merely a single sampled message is replaced by a resampled message in each hybrid transition allowing for a tighter reduction. Due to the hybrid's definition it will suffice to guess on vertex of C_k's neighbourhood fewer.

1.2 Previous Work

There are three not polynomially equivalent definitions of SO secure encryption (cf. [4]). Since messages in the IND-SO experiment have to be resampled conditioned on opened messages there are two notions based on indistinguishability: *Weak* IND-SO restricts to distributions that support *efficient conditional* resampling. [2] gave an indistinguishability-based notion for passive adversaries, usually referred to as IND-SO-CPA. *Full* IND-SO allows for arbitrary distributions on the messages and is due to [4] that adopted a notion for commitment schemes (cf. [2]) to encryption.

SIM-SO captures semantic security and demands that everything an adversary can output, can be computed by a simulator that only sees the messages of corrupted parties, whereas it does not see the public key, any ciphertext or any randomness. The notion dates back to [8] where the *selective decommitment* problem was studied and does not suffer from a distribution restriction as *weak* IND-SO since it does not involve resampling.

[2] gave the first IND-SO-CPA secure encryption scheme in the standard model based on lossy encryption. Selective-opening secure encryption can be constructed from *deniable encryption* (cf. [6]) as well as from *non-committing encryption* (cf. [7]). [3, 1] separated SIM-SO-CPA from IND-CPA security and showed that IND-CPA security entails *weak* IND-SO-CPA security if the messages are (basically) sampled independently. The same result was already established for commitment schemes in [8].

To date, it is the only positive result on IND-CPA implying *weak* IND-SO-CPA in the standard model. [4] separated *full* IND-CPA and SIM-SO-CPA security; neither of them implies the other. [10] proved that IND-CPA implies *weak* IND-SO-CPA in the generic group model for a certain class of encryption schemes and separated IND-CCA from *weak* IND-SO-CCA security.

Recently, Hofheinz et al. ([9]) managed to construct a contrived first IND-CPA (even IND-CCA) secure PKE that is not *weak* IND-SO-CPA secure. Thereby they have to rely on strong assumptions as the existence of *public-coin differing-inputs obfuscation* and certain *correlation intractable hash functions.*

2 Preliminaries

We denote the security parameter with λ. A function f is polynomial in n, $f(n) = \mathsf{poly}(n)$, if $f(n) = \mathcal{O}(n^c)$ for some $c > 0$. Let $0 < n := n(\lambda) = \mathsf{poly}(\lambda)$. A function $f(n)$ is negligible in n, $f(n) = \mathsf{negl}(n)$, if $f(n) = \mathcal{O}(n^{-c})$ for all $c > 0$. Any algorithm receives the unary representation 1^λ of the security parameter as first input. We say that an algorithm is a PPT algorithm, if it runs in probabilistic polynomial time (in λ). For a finite set $\mathcal{S}$ we denote the sampling of a uniform random element a by $a \leftarrow_\$ \mathcal{S}$, and the sampling according to some distribution $\mathfrak{D}$ by $a \leftarrow \mathfrak{D}$. For $a, b \in \mathbb{N}$, $a \leq b$ let $[a, b] := \{a, a+1, \ldots, b\}$ and $[a] := [1, a]$. For $a < b$ let $[b, a] := \emptyset$. For $\mathcal{I} \subseteq [n]$ let $\overline{\mathcal{I}} := [n] \setminus \mathcal{I}$. We use bold faced letters to denote vectors and vectors are of length n if not indicated otherwise. For a vector $\mathbf{m}$ and $i \in [n]$ let $\mathbf{m}_i$ denote the i^{th} entry of $\mathbf{m}$ and $|\mathbf{m}|$ the number of entries in $\mathbf{m}$. For a set $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\}$, $i_1 < \ldots < i_{|\mathcal{I}|}$ let $\mathbf{m}_\mathcal{I}$ denote the projection of $\mathbf{m}$ to its $\mathcal{I}$-entries: $\mathbf{m}_\mathcal{I} := (\mathbf{m}_{i_1}, \ldots, \mathbf{m}_{i_{|\mathcal{I}|}})$. For an event $\mathtt{E}$ let $\overline{\mathtt{E}}$ denote the complementary event.

2.1 Games

A game G is a collection of procedures/oracles $\{\textsc{Initialize}, \mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_t, \textsc{Finalize}\}$ for $t \geq 0$. Procedures P_1 to P_t and $\textsc{Finalize}$ might require some input parameters. We implicitly assume that boolean flags are initialized to false, numerical types are initialized to 0, sets are initialized to $\emptyset$, while strings are initialized to the empty string ϵ. An adversary $\mathcal{A}$ is *run in game* G if $\mathcal{A}$ calls $\textsc{Initialize}$. During the game $\mathcal{A}$ may run some procedure P_i as often as allowed by the game.

For each game in this paper, the "$\textsc{Open}$" procedure may be called an arbitrary number of times, while every other procedure is called one time during the games execution.

The game's interface is provided by the *challenger*. If $\mathcal{A}$ calls P, the output of P is returned to $\mathcal{A}$, except for the $\textsc{Finalize}$ procedure. On $\mathcal{A}$'s call of $\textsc{Finalize}$ the game ends and outputs whatever $\textsc{Finalize}$ returns. Let $\mathsf{G}^\mathcal{A} \Rightarrow \mathsf{out}$ denote the event that G runs $\mathcal{A}$ and outputs out. The *advantage* $\mathbf{Adv}(\mathsf{G}^\mathcal{A}, \mathsf{H}^\mathcal{A})$ of $\mathcal{A}$ in distinguishing games G and H is defined as $\left|\Pr[\mathsf{G}^\mathcal{A} \Rightarrow 1] - \Pr[\mathsf{H}^\mathcal{A} \Rightarrow 1]\right|$. Let $\mathtt{Bad}$ denote the event that a boolean flag $\mathtt{Bad}$ was set to *true* during the execution of some game.

2.2 Public-Key Encryption Schemes

A public-key encryption scheme consists of three PPT algorithms. Gen generates a key pair $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ on input 1^λ. The public key pk implicitly contains 1^λ and defines three finite sets: the message space $\mathcal{M}$, the randomness space $\mathcal{R}$, and the ciphertext space $\mathcal{C}$. Given pk, a message $m \in \mathcal{M}$ and randomness $r \in \mathcal{R}$, Enc outputs an encryption $c = \mathsf{Enc}_{pk}(m; r) \in \mathcal{C}$ of m under pk. The decryption algorithm Dec takes a secret key sk and a ciphertext $c \in \mathcal{C}$ as input, and outputs a message $m = \mathsf{Dec}_{sk}(c) \in \mathcal{M}$, or a special symbol $\perp \notin \mathcal{M}$ indicating that c is not a valid ciphertext. In the following let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ denote a public-key encryption scheme.

We require PKE to be correct: For all security parameters λ, for all $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, and for all $m \in \mathcal{M}$ we have $\Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m; r)) = m] = 1$ where the probability is taken over the choice of r. We apply Enc and Dec to message vectors $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$ and randomness $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$ as $\mathsf{Enc}(\mathbf{m}; \mathbf{r}) := (\mathsf{Enc}(\mathbf{m}_1; \mathbf{r}_1), \ldots, \mathsf{Enc}(\mathbf{m}_n; \mathbf{r}_n))$.

2.3 IND-CPA and mult-IND-CPA Security

We revise the standard security of IND-CPA security and give a definition of indistinguishable ciphertext vectors under chosen plaintext attacks that will allow for cleaner proofs of our results.

Definition 2.1 (mult-IND-CPA security) For PKE, an adversary $\mathcal{B}_{\mathsf{mult}}$, $s \in \mathbb{N}$ and a bit b we consider the $\mathsf{mult\text{-}IND\text{-}CPA}_{\mathsf{PKE},b}^{\mathcal{B}_{\mathsf{mult}}}$ game as given in Figure 1. $\mathcal{B}_{\mathsf{mult}}$ may only submit message vectors $\mathbf{m}^0$, $\mathbf{m}^1 \in \mathcal{M}^s$. To PKE, $\mathcal{B}_{\mathsf{mult}}$ and λ we associate the following advantage function

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda) := \mathbf{Adv}(\mathsf{mult\text{-}IND\text{-}CPA}_{\mathsf{PKE},0}^{\mathcal{B}_{\mathsf{mult}}}, \mathsf{mult\text{-}IND\text{-}CPA}_{\mathsf{PKE},1}^{\mathcal{B}_{\mathsf{mult}}}).$$

PKE is mult-IND-CPA secure if $\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda)$ is negligible for all PPT adversaries $\mathcal{B}_{\mathsf{mult}}$.

Procedure INITIALIZE	**Procedure** CHALLENGE $(\mathbf{m}^0, \mathbf{m}^1)$	**Procedure** FINALIZE(b')
$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$	$\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(\mathbf{m}^b)$	Return b'
Return pk	Return $\mathbf{c}$	

Figure 1: $\mathsf{mult\text{-}IND\text{-}CPA}_{\mathsf{PKE},b}$ game. $\mathcal{B}_{\mathsf{mult}}$ has to submit $\mathbf{m}^0, \mathbf{m}^1 \in \mathcal{M}^s$.

For an adversary $\mathcal{B}_{\mathsf{CPA}}$, we obtain the definition of IND-CPA security if we let $s = 1$ and write $\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda)$ instead of $\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda)$. A standard hybrid argument proves the following Lemma 2.2.

Lemma 2.2 *For any adversary $\mathcal{B}_{\mathsf{mult}}$ sending message vectors from $\mathcal{M}^s$ to the* mult-IND-CPA *game, there exists an* IND-CPA *adversary $\mathcal{B}_{\mathsf{CPA}}$ with roughly the same running time as $\mathcal{B}_{\mathsf{mult}}$ such that*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda) \leq s \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda).$$

2.4 IND-SO-CPA Security

In this section we recall an indistinguishability-based definition for selective opening security under chosen plaintext attacks and discuss the existing notions of SO security.

Definition 2.3 (efficiently resamplable distribution) Let $\mathcal{M}$ be a finite set. A family of distributions $\{\mathfrak{D}_\lambda\}_{\lambda \in \mathbb{N}}$ over $\mathcal{M}^n = \mathcal{M}^{n(\lambda)}$ is *efficiently resamplable* if the following properties hold for every $\lambda \in \mathbb{N}$:

length-consistency. For every $i \in [n]$ we have $\Pr_{\mathbf{m}^1 \leftarrow \mathfrak{D}_\lambda, \mathbf{m}^2 \leftarrow \mathfrak{D}_\lambda}[|\mathbf{m}_i^1| = |\mathbf{m}_i^2|] = 1$.

resamplability. There exists a PPT resampling algorithm $\mathsf{Resamp}_{\mathfrak{D}_\lambda}(\cdot, \cdot)$ that runs on $(\mathbf{m}, \mathcal{I})$ for $\mathbf{m} \in \mathcal{M}^n$, $\mathcal{I} \subseteq [n]$ and outputs a $\mathfrak{D}_\lambda$-distributed vector $\mathbf{m}' \in \mathcal{M}^n$ conditioned on $\mathbf{m}'_{\mathcal{I}} = \mathbf{m}_{\mathcal{I}}$.

A class of families of distributions $\mathcal{D}$ is efficiently resamplable if every family $\{\mathfrak{D}_\lambda\}_{\lambda \in \mathbb{N}} \in \mathcal{D}$ is efficiently resamplable.

Since the security parameter uniquely specifies an element of a family $\mathfrak{D}_\lambda$ we write $\mathfrak{D}$ instead of $\mathfrak{D}_\lambda$ whenever the security parameter is already fixed.

Definition 2.4 For PKE, a bit b, an adversary $\mathcal{A}_{\mathsf{SO}}$ and a class of families of distributions $\mathcal{D}$ over $\mathcal{M}^n$ we consider the $\mathsf{IND\text{-}SO\text{-}CPA}^{\mathcal{A}_{\mathsf{SO}}}_{\mathsf{PKE},b}$ game in Figure 2. Run in the game, $\mathcal{A}_{\mathsf{SO}}$ calls ENC once right after INITIALIZE and has to submit $\mathfrak{D} \in \mathcal{D}$ along with a PPT resampling algorithm $\mathsf{Resamp}_{\mathfrak{D}}$. $\mathcal{A}_{\mathsf{SO}}$ may call OPEN multiple times and invokes CHALLENGE once after its last OPEN query before calling FINALIZE. We define the advantage of $\mathcal{A}_{\mathsf{SO}}$ run in the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},b}$ game as

$$\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) := \mathbf{Adv}(\mathsf{IND\text{-}SO\text{-}CPA}^{\mathcal{A}_{\mathsf{SO}}}_{\mathsf{PKE},0}, \mathsf{IND\text{-}SO\text{-}CPA}^{\mathcal{A}_{\mathsf{SO}}}_{\mathsf{PKE},1}).$$

PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure w.r.t.* $\mathcal{D}$ if $\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda)$ is negligible for all PPT $\mathcal{A}_{\mathsf{SO}}$.

Procedure INITIALIZE $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ Return pk	**Procedure** OPEN(i) $\mathcal{I} := \mathcal{I} \cup \{i\}$ Return $(\mathbf{m}^0_i, \mathbf{r}_i)$
Procedure ENC$(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})$ $\mathbf{m}^0 \leftarrow \mathfrak{D}$ $\mathbf{r} \leftarrow_\$ \mathcal{R}^n$ $\mathbf{c} = \mathsf{Enc}_{pk}(\mathbf{m}^0; \mathbf{r})$ Return $\mathbf{c}$	**Procedure** CHALLENGE $\mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I})$ Return $\mathbf{m}^b$ **Procedure** FINALIZE(b') Return b'

Figure 2: $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},b}$ game.

NOTIONS OF SELECTIVE OPENING SECURITY. Definition 2.4 is in the spirit of [2] and applies the same changes as in [4]. [4] renamed $\mathsf{IND\text{-}SO\text{-}CPA}$ to *weak* $\mathsf{IND\text{-}SO\text{-}CPA}$ and introduced a strictly stronger definition than $\mathsf{IND\text{-}SO\text{-}CPA}$ called *full* $\mathsf{IND\text{-}SO\text{-}CPA}$, where $\mathcal{A}_{\mathsf{SO}}$ may submit any distribution (even a not efficiently resamplable) and does not provide a resampling algorithm. We refer to the security notion given in definition 2.4 as $\mathsf{IND\text{-}SO\text{-}CPA}$ security.

3 Selective Opening for Graph-Induced Distributions

This Section considers graph-induced distributions and identifies connectivity properties such that $\mathsf{IND\text{-}CPA}$ entails $\mathsf{IND\text{-}SO\text{-}CPA}$ security. We introduce some notation in Section 3.1. Sections 3.2 and 3.4 discuss a hybrid argument that contemplates the connected components of $G_{\overline{\mathcal{I}}}$, switching one of them from *sampled* to *resampled* in each transition. Section 3.5 will discuss a different hybrid argument that will allow for tighter proofs if the distribution-inducing graph is acyclic.

3.1 Graphs

A *directed graph* G consists of a set of vertices V, identified with $[n]$ for $n > 0$ and a set of edges $E \subseteq V^2 \setminus \{(v, v) : v \in V\}$, i.e. we do not allow for loops. G is *undirected* if $(v_2, v_1) \in E$ for each $(v_1, v_2) \in E$. For $V' \subseteq V$ let $G_{V'} := (V', E')$ denote the *induced subgraph* of G where $E' := E \cap V'^2$. For $G = (V, E)$ we obtain its *undirected version*, $G^{\leftrightarrow} = (E^{\leftrightarrow}, V)$ where $E \subseteq E^{\leftrightarrow}$, by adding the minimum number of edges to E such that the graph becomes undirected. For $V' \subseteq V$ let $N(V') := \{v \in V \setminus V' : \exists v' \in V' \; s.t. \; (v, v') \in E^{\leftrightarrow}\}$ denote the *(open) neighbourhood* of V' in G. For a vertex v, we denote with $P(v) = \{j : (j, v) \in E)$ the set of its *parents*.

A *path* from v_1 to v_ℓ in G is a list of at least two vertices $(v_1, \ldots, v_\ell)$ where $v_i \in V$ for $i \in [\ell]$ and $(v_i, v_{i+1}) \in E$ for all $i \in [\ell - 1]$. If there is a path from u to v, u is a *predecessor* of v. Let $\mathsf{pred}(v)$ denote the set of all predecessors of v. A *cycle* is a path where $v_\ell = v_1$. If G contains no cycles, it is *acyclic*. A directed, acyclic graph is called DAG.

A non-empty subset $V' \subseteq V$ is *connected* in G if for every distinct pair of vertices $(v_1, v_2) \in V'$ there exists a path form v_1 to v_2 in $G^{\leftrightarrow}$. *G is connected* if V is connected in G. G is *disconnected* if G is not connected. We assume G to be connected if not stated otherwise. A (set-)maximal connected set of vertices of G is called *connected component*.

Notational Convention. We do not distinguish between the i^{th} message of a n-message vector and vertex i in a graph on n vertices.

Definition 3.1 (graph-induced distribution) Let $\{\mathfrak{D}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of distributions over $\mathcal{M}^n$ and let $\{G_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of not necessarily connected graphs on n vertices. We say that $\{\mathfrak{D}_\lambda\}_{\lambda\in\mathbb{N}}$ is $\{G_\lambda\}_{\lambda\in\mathbb{N}}$*-induced* if for all $\lambda \in \mathbb{N}$, all $i \in [n]$ and $\mathbf{M} = (\mathbf{M}_1, \ldots, \mathbf{M}_n) \leftarrow \mathfrak{D}_\lambda$ the distribution of $\mathbf{M}_j$, given all its predecessors in G_λ, actually depends on its parents in G_λ only. That is, for all $\lambda \in \mathbb{N}$ and $i \in [n]$ we have

$$\Pr_{\mathbf{M}\leftarrow\mathfrak{D}_\lambda}\Big[\mathbf{M}_j = \mathbf{m}_j \Big| \bigwedge_{i\in\mathsf{pred}(j)} \mathbf{M}_i = \mathbf{m}_i\Big] = \Pr_{\mathbf{M}\leftarrow\mathfrak{D}_\lambda}\Big[\mathbf{M}_j = \mathbf{m}_j \Big| \bigwedge_{i\in P(j)} \mathbf{M}_i = \mathbf{m}_i\Big].$$

We demand that for any $\lambda \in \mathbb{N}$ one can efficiently reconstruct G_λ from $\mathfrak{D}_\lambda$.

As with a family of distributions, we drop the security parameter and say that *$\mathfrak{D}$ is G-induced* whenever λ is already fixed. Note that G may contain cycles and may be undirected.

Even though our proof ideas can be applied to disconnected graphs directly, Sections 3.2, 3.4, 3.5 consider *connected graphs* for simplicity. A hybrid argument over the connected components of a graph as given in Section 3.6 extends any result to disconnected graphs.

3.2 A bound using connected subgraphs

Definition 3.2 (number of connected subgraphs) Let $G = (V, E)$. We define the *number of connected subgraphs* of G:

$$S(G) := \left|\{V' \subseteq V : V' \text{ connected}\}\right|.$$

For example, for a chain graph on n vertices we have $S(G) = \frac{1}{2} \cdot n \cdot (n+1)$ and for the complete graph C_n on n vertices we have $S(C_n) = 2^n - 1$.

Theorem 3.3 *Let* PKE *be* $\mathsf{IND\text{-}CPA}$ *secure, then* PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure w.r.t. the class of efficiently resamplable and G-induced distribution families over $\mathcal{M}^n$ where $S(G) = \mathsf{poly}(n)$ and G is connected.*

Precisely, for any adversary $\mathcal{A}_{\mathsf{SO}}$ run in the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE}}$ *game, there exists an* $\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$ *adversary $\mathcal{B}_{\mathsf{CPA}}$ with roughly the same running time as $\mathcal{A}_{\mathsf{SO}}$ such that*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \frac{1}{2} \cdot n \cdot (n+1) \cdot S(G_\lambda) \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda).$$

Proof Idea. Recall the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},b}$ game given in Figure 2. During Challenge the game sends $\mathbf{m}^b$, where $\mathbf{m}^0_{\overline{\mathcal{I}}}$ consists of sampled messages, while $\mathbf{m}^1_{\overline{\mathcal{I}}}$ is resampled (conditioned on $\mathbf{m}^1_{\mathcal{I}} = \mathbf{m}^0_{\mathcal{I}}$) otherwise. We will define hybrid games $\mathsf{H}_0, \mathsf{H}_1, \ldots, \mathsf{H}_n$. For this, let $\mathcal{S} \subseteq 2^V$ denote all the connected subgraphs of G. We have $|\mathcal{S}| = S(G)$.

Note that $G_{\overline{\mathcal{I}}}$ consists of connected components $C_1, \ldots, C_{n'} \subseteq \mathcal{S}$ for some $n' \leq (n+1)/2$. We assume those components to be ordered (e.g. by the smallest vertex contained in each).

Thus, if $b = 1$ in the $\mathsf{IND\text{-}SO\text{-}CPA}$ game, then the challenger can resample $\mathbf{m}^0_{\overline{\mathcal{I}}}$ in n' batches $\mathbf{m}^1_{C_1}, \ldots, \mathbf{m}^1_{C_{n'}}$ (as $\overline{\mathcal{I}} = \cup_{i=1}^{n'} C_i$). Moreover, each batch can be resampled *independently* (i.e., just as a function of $\mathbf{m}^0_{\mathcal{I}}$ and $\mathfrak{D}$).

Proof of Theorem 3.3: For $k = 0, \ldots, n$ we define hybrid game H_k as the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE}}$ game, whereby the messages of the first k batches $C_1, \ldots, C_k$ are resampled during Challenge while the remaining batches stay sampled.

Every procedure except Challenge remains as in Definition 2.4, while the Challenge procedure is given in Figure 3.

Procedure Challenge

$\mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I})$

$$\mathbf{m}_i := \begin{cases} \mathbf{m}^1_i & \text{for } i \in \bigcup_{j=1}^{k} C_j \\ \mathbf{m}^0_i & \text{else} \end{cases}$$

Return $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$

Figure 3: Challenge procedure of hybrid game H_k. C_i denotes the i^{th} connected component of $G_{\overline{\mathcal{I}}}$. The challenge vector contains resampled messages in the first k batches $C_1, \ldots, C_k$ while the other messages remain sampled.

So, in H_k the first k batches of the messages that $\mathcal{A}_{\mathsf{SO}}$ gets are resampled. Clearly, H_0 is the (real) $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},0}$ game and $\mathsf{H}_{n'}$ for some $n' \leq (n+1)/2$ is the (random) $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},1}$ game. Note that for $k, j \in [n', n]$ hybrids H_k and H_j are identical. Clearly

$$\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) = \mathbf{Adv}(\mathsf{H}_0^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{n'}^{\mathcal{A}_{\mathsf{SO}}}) \leq \sum_{k=0}^{n'-1} \mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}}).$$

We upper-bound the distance of two consecutive hybrids using Lemma 3.4.

Lemma 3.4 *For every adversary $\mathcal{A}_{\mathsf{SO}}$ that distinguishes hybrids H_k and H_{k+1}, there exists a* $\mathsf{mult\text{-}IND\text{-}CPA}$ *adversary $\mathcal{B}_{\mathsf{mult}}$ with roughly the same running time such that*

$$\mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}}) \leq S(G) \cdot \mathbf{Adv}^{\mathsf{mult\text{-}IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{mult}}, \lambda).$$

Proof of Lemma 3.4: We construct adversary $\mathcal{B}_{\mathsf{mult}}$ as follows (cf. Figure 4):

$\mathcal{B}_{\mathsf{mult}}$ forwards pk to $\mathcal{A}_{\mathsf{SO}}$ and picks $C^*_{k+1} \leftarrow_\$ \mathcal{S}$ uniformly random (trying to guess C_{k+1}) after receiving $(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})$. $\mathcal{B}_{\mathsf{mult}}$ samples $\mathbf{m}^0 \leftarrow \mathfrak{D}$ and resamples $\mathbf{m}^1$ keeping the neighbourhood of C^*_{k+1} fixed. $\mathcal{B}_{\mathsf{mult}}$ submits $(\mathbf{m}^0_{C^*_{k+1}}, \mathbf{m}^1_{C^*_{k+1}})$ to its $\mathsf{mult\text{-}IND\text{-}CPA}$ challenger, obtains ciphertexts for positions in C^*_{k+1}, picks randomness and encrypts each message in $\overline{C^*_{k+1}}$ on its own. $\mathcal{B}_{\mathsf{mult}}$

Procedure $\textsc{Initialize}$

$pk \leftarrow \textsc{Initialize}_{\textsf{mult-IND-CPA}}(1^\lambda)$

Return pk

Procedure $\textsc{Enc}(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})$

$C^*_{k+1} \leftarrow_\$ \mathcal{S}$

$\mathbf{m}^0 \leftarrow \mathfrak{D}$

$\mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, N(C^*_{k+1}))$

$\mathbf{c}_{C^*_{k+1}} \leftarrow \textsc{Challenge}_{\textsf{mult-IND-CPA}}(\mathbf{m}^0_{C^*_{k+1}}, \mathbf{m}^1_{C^*_{k+1}})$

$\mathbf{r} \leftarrow_\$ \mathcal{R}^n$

$\mathbf{c}_i = \begin{cases} \mathbf{c}_i & \text{for } i \in C^*_{k+1} \\ \mathsf{Enc}_{pk}(\mathbf{m}^0_i; \mathbf{r}_i) & \text{else} \end{cases}$

Return $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n)$

Procedure $\textsc{Open}(i)$

if $i \in C^*_{k+1}$

  $\mathtt{Bad} := \mathit{true}$

$\mathcal{I} := \mathcal{I} \cup \{i\}$

Return $(\mathbf{m}^0_i, \mathbf{r}_i)$

Procedure $\textsc{Challenge}$

if $C^*_{k+1} \neq C_{k+1}$

  $\mathtt{Bad} := \mathit{true}$

$\widetilde{\mathbf{m}}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I})$

$\mathbf{m}_i = \begin{cases} \widetilde{\mathbf{m}}^1_i & \text{for } i \in \cup_{j=1}^k C_j \\ \mathbf{m}^0_i & \text{else} \end{cases}$

Return $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$

Procedure $\textsc{Finalize}(b')$

$\textsc{Finalize}_{\textsf{mult-IND-CPA}}(b')$

Figure 4: $\mathcal{A}_{\mathsf{SO}}$'s game interface as provided by $\mathcal{B}_{\mathsf{mult}}$ run in the mult-IND-CPA game. $\mathcal{B}_{\mathsf{mult}}$ interpolates between hybrids H_k, H_{k+1} for $k \in [0, n-1]$.

sends $(\mathbf{c}_1, \ldots, \mathbf{c}_n)$ to $\mathcal{A}_{\mathsf{SO}}$, embedding its challenge at positions C^*_{k+1} and answers opening queries honestly if they do not occur on C^*_{k+1}. If $\mathcal{A}_{\mathsf{SO}}$ issues such a query, $\mathcal{B}_{\mathsf{mult}}$ cannot answer and sets $\mathtt{Bad} := \mathit{true}$ since it guessed C_{k+1} wrong. During $\textsc{Challenge}$, $\mathcal{B}_{\mathsf{mult}}$ verifies that it guessed C_{k+1} correctly and sets $\mathtt{Bad} := \mathit{true}$ if not. $\mathcal{B}_{\mathsf{mult}}$ resamples messages $\widetilde{\mathbf{m}}^1$ that are sent in the first k batches while messages from $\mathbf{m}^0$ are sent in every other position. $\mathcal{B}_{\mathsf{mult}}$ outputs $\mathcal{A}_{\mathsf{SO}}$'s output.

Assume, $\mathcal{B}_{\mathsf{mult}}$ guessed correctly, i.e. $C^*_{k+1} = C_{k+1}$. Clearly, $\mathcal{B}_{\mathsf{mult}}$ perfectly simulates hybrids H_k and H_{k+1} for messages and ciphertexts at positions in $\overline{C_{k+1}}$. Run in $\mathsf{mult\text{-}IND\text{-}CPA}_{\mathsf{PKE},0}$, $\mathcal{B}_{\mathsf{mult}}$ obtains $\mathsf{Enc}_{pk}(\mathbf{m}^0_{C_{k+1}})$ and $\mathcal{A}_{\mathsf{SO}}$ receives encryptions of sampled messages. During $\textsc{Challenge}$ the $k+1^{th}$ batch contains sampled messages $\mathbf{m}^0_{C_{k+1}}$, thus, $\mathcal{B}_{\mathsf{mult}}$ perfectly simulates hybrid H_k.

When $\mathcal{B}_{\mathsf{mult}}$ is run in $\mathsf{mult\text{-}IND\text{-}CPA}_{\mathsf{PKE},1}$, $\mathcal{A}_{\mathsf{SO}}$ obtains encryptions of *resampled* messages $\mathsf{Enc}_{pk}(\mathbf{m}^1_{C_{k+1}})$ while it expects encrypted *sampled* messages: $\mathsf{Enc}_{pk}(\mathbf{m}^0_{C_{k+1}})$. During $\textsc{Challenge}$ $\mathcal{A}_{\mathsf{SO}}$ expects *resampled* messages $\widetilde{\mathbf{m}}^1_{C_{k+1}}$ but obtains *sampled* $\mathbf{m}^0_{C_{k+1}}$. Thus, the *sampled* and *resampled* messages change roles on C_{k+1}.

However, $\mathbf{m}^0_{C_{k+1}} \equiv \mathbf{m}^1_{C_{k+1}}$ since the messages in $N(C_{k+1})$ were fixed when resampling $\mathbf{m}^1$ and the distribution of messages in C_{k+1} depends on $\mathfrak{D}$ and messages in positions $N(C_{k+1})$ only.

Additionally, $\mathbf{m}^1_{C_{k+1}} \equiv \widetilde{\mathbf{m}}^1_{C_{k+1}}$ for $\mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, N(C_{k+1}))$ and $\widetilde{\mathbf{m}}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I})$ since the distribution of messages in C_{k+1} solely depends on $\mathfrak{D}$ and messages in $N(C_{k+1}) \subseteq \mathcal{I}$ and $\mathcal{A}_{\mathsf{SO}}$'s view is identical to hybrid H_{k+1}. We have

$$\Pr[\mathsf{mult\text{-}IND\text{-}CPA}^{\mathcal{B}_{\mathsf{mult}}}_{\mathsf{PKE},0} \Rightarrow 1] = \Pr[\mathsf{H}^{\mathcal{A}_{\mathsf{SO}}}_k \Rightarrow 1 \wedge \overline{\mathtt{Bad}}]$$

$$\Pr[\mathsf{mult\text{-}IND\text{-}CPA}^{\mathcal{B}_{\mathsf{mult}}}_{\mathsf{PKE},1} \Rightarrow 1] = \Pr[\mathsf{H}^{\mathcal{A}_{\mathsf{SO}}}_{k+1} \Rightarrow 1 \wedge \overline{\mathtt{Bad}}].$$

Observe that Bad does not happen when $\mathcal{B}_{\mathsf{mult}}$ guessed C_{k+1} correctly. Since $\overline{\mathsf{Bad}}$ is independent of $\mathcal{A}_{\mathsf{SO}}$'s output in a hybrid and $|\mathcal{S}| = S(G)$ we have

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda) \geq \frac{1}{S(G)} \cdot \mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}})$$

to conclude the proof. ∎

We proceed with the proof of Theorem 3.3. Using Lemma 3.4:

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \sum_{k=0}^{n'-1} \mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}}) \leq \sum_{k=0}^{n'-1} S(G_\lambda) \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda).$$

$\mathcal{B}_{\mathsf{mult}}$ sends message vectors of length $|C_{k+1}^*| \leq n$ to its $\mathsf{mult\text{-}IND\text{-}CPA}$ challenger. Using Lemma 2.2:

$$\leq \sum_{k=0}^{n'-1} n \cdot S(G_\lambda) \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda) \leq \frac{1}{2} \cdot n \cdot (n+1) \cdot S(G_\lambda) \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda)$$

since $n' \leq (n+1)/2$ to complete the proof of Theorem 3.3. ∎

3.3 Markov Distributions

Definition 3.5 Let $\{\mathfrak{D}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of distributions over $\mathcal{M}^n$. Let $\mathbf{M} = (\mathbf{M}_1, \ldots, \mathbf{M}_n)$ denote a vector of $\mathcal{M}$-valued random variables. We say $\{\mathfrak{D}_\lambda\}_{\lambda\in\mathbb{N}}$ is *Markov* if the following holds for all $\lambda \in \mathbb{N}$ and all $\mathbf{M}_i$:

$$\Pr_{\mathbf{M}\leftarrow\mathfrak{D}_\lambda}[\mathbf{M}_i = \mathbf{m}_i | \bigwedge_{j=i+1}^{n} \mathbf{M}_j = \mathbf{m}_j] = \Pr_{\mathbf{M}\leftarrow\mathfrak{D}_\lambda}[\mathbf{M}_i = \mathbf{m}_i | \mathbf{M}_{i+1} = \mathbf{m}_{i+1}].$$

Definition 3.5 matches the "usual" definition of a Markov distribution but we impose the Markov property on $(\mathbf{M}_n, \ldots, \mathbf{M}_1)$ instead of $(\mathbf{M}_1, \ldots, \mathbf{M}_n)$ to ease later notation.

Note that Markov distributions can be seen as graph-induced distributions where the graph $G = (V, E)$ is a chain on n vertices: $V = [n]$, $E = \{(i, i-1) : i \in [n]\}$. Since $S(G) = \frac{1}{2} \cdot n \cdot (n+1)$ we immediately obtain Corollary 3.6 whose proof directly follows from Theorem 3.3.

Corollary 3.6 *Let* PKE *be* $\mathsf{IND\text{-}CPA}$ *secure , then* PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure w.r.t. efficiently resamplable Markov distributions over* $\mathcal{M}^n$*.*

Precisely, for any adversary $\mathcal{A}_{\mathsf{SO}}$ *run in the* $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE}}$ *game, there exists an* $\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$ *adversary* $\mathcal{B}_{\mathsf{CPA}}$ *with roughly the same running time as* $\mathcal{A}_{\mathsf{SO}}$ *such that*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \frac{1}{4} \cdot n^2 \cdot (n+1)^2 \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda).$$

3.4 A bound using the maximum border

Definition 3.7 (maximum border) Let $G = (V, E)$. We define the *maximum border* of G as the maximum over the size of the neighbourhood of every connected subgraph in G.

$$B(G) := \max\{|N(V')| \;:\; V' \subseteq V \text{ connected}\}.$$

For example, if G is an n-path for $n \geq 3$ then $B(G) = 2$. For the complete graph or star graph on n vertices we have $B(G) = n - 1$. Notice that $B(G) < n$.

During the reduction in Section 3.2 we guessed a connected component in $G_{\overline{\mathcal{I}}}$ that would be switched from sampled to resampled in a hybrid transition. Alternatively, we can guess a connected component in $G_{\overline{\mathcal{I}}}$ via its neighbourhood. The following Theorem 3.8 expresses $S(G)$ in terms of $B(G)$.

Theorem 3.8 *Let G be a connected graph. Then the following bound on $S(G)$ holds:*

$$S(G) \leq \frac{2}{(B(G)-1)!} \cdot n^{B(G)} \quad \textit{for all} \quad 0 < B(G) \leq \frac{n-2}{3}.$$

We begin with a simple observation given in Lemma 3.9 before proving Theorem 3.8.

Lemma 3.9 *Let $G = (V, E)$ and $V_1 \neq V_2$ each of them connected in G such that $N(V_1) = N(V_2)$. Then $V_1 \cap V_2 = \emptyset$.*

Proof of Lemma 3.9: Assume $V_1 \cap V_2 \neq \emptyset$. As $V_1 \neq V_2$ we have $V_1 \setminus V_2 \neq \emptyset$ without loss of generality. Because V_1 is connected, there exist vertices $v_\cap \in V_1 \cap V_2$ and $v_1 \in V_1 \setminus V_2$ such that $(v_1, v_\cap) \in E$. Since $v_1 \notin V_2$, $v_\cap \in V_2$ and $(v_1, v_\cap) \in E$ we see that $v_1 \in N(V_2)$. As $N(V_2) = N(V_1)$ it follows that $v_1 \in N(V_1)$; a contradiction by definition of $N(V_1)$. ■

Proof of Theorem 3.8: Let $B := B(G)$. We have

$$S(G) = \sum_{i=0}^{B} \left|\left\{V' \subseteq V \colon V' \text{ connected} \wedge |N(V')| = i\right\}\right|$$

For $i = 0$ we count the connected components of G.

$$\begin{aligned} &= 1 + \sum_{i=1}^{B} \left|\left\{V' \subseteq V \colon V' \text{ connected} \wedge |N(V')| = i\right\}\right| \\ &= 1 + \sum_{i=1}^{B} \sum_{\substack{V_i \subseteq V \\ |V_i| = i}} \left|\left\{V' \subseteq V \colon V' \text{ connected} \wedge N(V') = V_i\right\}\right| \end{aligned}$$

Let $V_i \subseteq V$ be a non-empty subset of V and $\{V' \subseteq V \colon V' \text{ connected} \wedge N(V') = V_i\} = \{V'_1, \ldots, V'_k\}$ for appropriate k. Applying Lemma 3.9 to $V'_1, \ldots, V'_k$, we see that those sets V'_j are pairwise disjoint. Fix any vertex $v_i \in V_i = N(V'_i)$. Since $N(V'_j) = V_i$ for $j \in [k]$ and all V'_j are pairwise disjoint, there exists at least one vertex v'_j in each V'_j such that $(v'_j, v_i) \in E$ for all $j \in [k]$. Thus, $N(v_i) \geq k$, i.e. $B \geq k$. Hence, $k \leq B$ for given B and we obtain an upper bound for the number of possible sets V' for each fixed V_i. It follows

$$S(G) \leq 1 + \sum_{i=1}^{B} \sum_{\substack{V_i \subseteq V \\ |V_i| = i}} B = 1 + B \cdot \sum_{i=1}^{B} \binom{n}{i} \leq B \cdot \sum_{i=0}^{B} \binom{n}{i}. \tag{1}$$

To bound the sum in (1) we use the geometric series and want to upper-bound[1] the quotient of two consecutive binomial coefficients by $\frac{1}{2}$:

$$\frac{\binom{n}{i}}{\binom{n}{i+1}} = \frac{i+1}{n-i} \leq \frac{1}{2} \Leftrightarrow i \leq \frac{n-2}{3}.$$

Hence

$$B \cdot \sum_{i=0}^{B} \binom{n}{i} \leq B \cdot \sum_{i=0}^{B} \frac{1}{2^i} \binom{n}{B} \leq B \cdot \binom{n}{B} \cdot \sum_{i=0}^{\infty} \frac{1}{2^i} \leq 2 \cdot B \cdot \frac{n^B}{B!} = \frac{2}{(B-1)!} \cdot n^B$$

for $B(G) \leq \frac{n-2}{3}$ to conclude the proof. ∎

Since Corollary 3.10 ensures polynomial loss in the reduction for $B(G) = \mathsf{const}.$ and we are interested in asymptotic statements, we do not consider the restriction to $n \geq 3 \cdot B(G) + 2$ grave. One can easily obtain a version of Theorem 3.8 that is weaker by a factor of roughly $B(G)$ but holds for all $B(G) < n$. To this end one bounds the sum of binomial coefficients in (1) in terms of the incomplete upper gamma function Γ to get

$$\sum_{i=1}^{B} \binom{n}{i} \leq \sum_{i=1}^{B} \frac{n^i}{i!} = \frac{e^n \Gamma(B+1, n)}{B!} - 1.$$

Using a nice bound on Γ due to [11] that can be found in [5] we obtain a bound for $B(G) < n$. We obtain the following Corollary 3.10.

Corollary 3.10 *Let* PKE *be* $\mathsf{IND\text{-}CPA}$ *secure, then* PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure w.r.t. the class of efficiently resamplable and G-induced distribution families over $\mathcal{M}^n$ where $B(G) = \mathsf{const.}$, $n \geq 3 \cdot B(G) + 2$ and G is connected.*

Concretely, for any adversary $\mathcal{A}_{\mathsf{SO}}$ in the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE}}$ game, there exists an $\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$ adversary $\mathcal{B}_{\mathsf{CPA}}$ with roughly the same running time as $\mathcal{A}_{\mathsf{SO}}$ such that

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \frac{n+1}{(B(G_\lambda)-1)!} \cdot n^{B(G_\lambda)+1} \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda).$$

Proof of Corollary 3.10: The proof directly follows from Theorem 3.3 and Theorem 3.8. ∎

Think of a direct reduction for proving Corollary 3.10 as implicitly guessing C_{k+1} via guessing $N(C_{k+1})$ by picking up to $B(G)$ vertices in G and guessing on of at most $B(G)$ connected subgraphs that has the guessed neighbourhood.

Note that Corollary 3.10 cannot provide a tighter bound on the loss than Theorem 3.3. In particular, there is (even) connected graphs where Theorem 3.3 ensures an at most polynomial loss, while Corollary 3.10 does not. For instance, let G be the star graph on $\log n$ vertices attached to the chain graph of $n - \log n$ vertices, then $S(G) = \mathsf{poly}(n)$, but $B(G) > \mathsf{const}$.

[1]It suffices to strictly upper-bound by 1 to ensure convergence.

3.5 A Tighter Reduction for Acyclic Graphs

While we consider graph-induced distributions for arbitrary graphs in Sections 3.2 - 3.4 we consider DAG-induced distributions in Section 3.5 where we obtain a tighter reduction than presented in Corollary 3.10.

For a DAG G, we require that the vertices are semi-ordered in such a way that there is no directed path from i to j for $i < j$. Note that such an ordering always exists as G has no cycles.

Theorem 3.11 *Let* PKE *be* $\mathsf{IND\text{-}CPA}$ *secure, then* PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure w.r.t. the class of efficiently resamplable and* G*-induced distribution families over* $\mathcal{M}^n$ *where* $B(G) = \mathsf{const.}$ *and* G *is a connected* DAG.

Precisely, for any adversary $\mathcal{A}_{\mathsf{SO}}$ *run in the* $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE}}$ *game, there exists an* $\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$ *adversary* $\mathcal{B}_{\mathsf{CPA}}$ *with roughly the same running time as* $\mathcal{A}_{\mathsf{SO}}$ *such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq 3 \cdot n^{B(G_\lambda)+1} \cdot \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{CPA}}, \lambda).$$

Proof of Theorem 3.11: We proceed in a sequence of hybrid games $\mathsf{H}_0, \mathsf{H}_1, \ldots, \mathsf{H}_n$ and switch message $\mathbf{m}_{k+1}$ from sampled to resampled in hybrid transition H_k to H_{k+1}. Hybrid H_k will return the sampled messages for all positions $[k+1, n] \cup \mathcal{I}$, but resampled messages on all positions $[k] \setminus \mathcal{I}$ where the resampling is conditioned on *every message in* $[k+1, n] \cup \mathcal{I}$. The code for CHALLENGE in given in Figure 5, every other procedure stays as in Figure 2.

<u>**Procedure**</u> <u>CHALLENGE</u>
$\mathbf{m} \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, [k+1, n] \cup \mathcal{I})$
Return $\mathbf{m}$

Figure 5: CHALLENGE procedure of hybrid game H_k. For $k = n$ we have $[n+1, n] = \emptyset$.

Hybrid H_0 (resp. H_n) is identical to the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},0}$ game (resp. $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},1}$), hence

$$\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) = \mathbf{Adv}(\mathsf{H}_0^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_n^{\mathcal{A}_{\mathsf{SO}}}) \leq \sum_{k=0}^{n-1} \mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}}).$$

We bound the distance of two consecutive hybrids H_k, H_{k+1} and proceed with the following lemma.

Lemma 3.12 *For every adversary* $\mathcal{A}_{\mathsf{SO}}$ *that distinguishes hybrids* H_k *and* H_{k+1} *there exists a* $\mathsf{mult\text{-}IND\text{-}CPA}$ *adversary* $\mathcal{B}_{\mathsf{mult}}$ *with roughly the same running time such that*

$$\mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}}) \leq \Pr[\overline{\mathsf{Bad}_k}]^{-1} \cdot \mathbf{Adv}^{\mathsf{mult\text{-}IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{mult}}, \lambda)$$

where $\Pr[\overline{\mathsf{Bad}_k}]^{-1} = \sum_{i=0}^{B(G_\lambda)-1} \binom{k}{i}$ *for* $k < n-1$ *and* $\Pr[\overline{\mathsf{Bad}_k}]^{-1} = \sum_{i=0}^{B(G_\lambda)} \binom{k}{i}$ *for* $k = n-1$.

PROOF IDEA: We construct a $\mathsf{mult\text{-}IND\text{-}CPA}$ adversary $\mathcal{B}_{\mathsf{mult}}$ that interpolates between hybrids H_k and H_{k+1}. Ideally $\mathcal{B}_{\mathsf{mult}}$ embeds its own challenge at position $k+1$, but might have to resample some already resampled messages in $\mathbf{m}_{[k]}$ to avoid inconsistencies. Let middle denote the connected component in $G_{[k+1] \setminus \mathcal{I}}$ that contains $\mathbf{m}_{k+1}$. Let $\mathsf{right} := [k+2, n]$, and $\mathsf{left} := \overline{(\mathsf{middle} \cup \mathsf{right})}$.

Observe that it is sufficient to resample middle again to obtain consistent resampled messages. In particular, there is no need to resample any right message due to the semi-order imposed on the vertices, as a message in right does not depend on any message in $\overline{\mathsf{right}}$ (cf. Figure 6). The reduction will guess middle to embed its mult-IND-CPA challenge, while it waits for all opening queries to happen to resample the left messages. Note that middle and left are disconnected in $G_{\overline{\mathcal{I}}}$, thus can be resampled independently of each other only depending on their respective neighbourhood. Since right messages are fixed while resampling, it suffices to guess $N(\mathsf{middle}) \cap [k]$. Further, G is connected, i.e. $N(\mathsf{middle})$ contains at least one vertex from $\mathsf{right} = [k+2, n]$ as long as $k < n-1$. Hence, for $k < n-1$, we have $|N(\mathsf{middle}) \cap [k]| \leq B(G) - 1$.

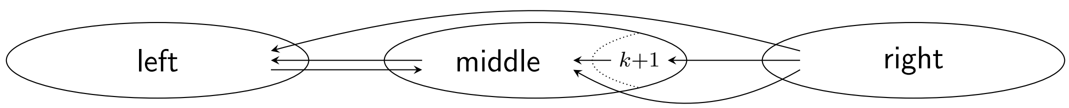

Figure 6: Structure of G. Edges between particular sets cannot exist if there is no arrow depicted. If $\mathsf{right} \neq \emptyset$, there is at least one edge from right to middle since G is connected. left and middle are disconnected in $G_{\overline{\mathcal{I}}}$.

Proof of Lemma 3.12: For $k \in [0, n]$ and $i \in [n]$ let $\mathtt{Open}_k(i)$ denote the event that $\mathcal{A}_{\mathsf{SO}}$ calls $\textsc{Open}(i)$ in hybrid H_k. Two arbitrary hybrids only differ in the $\textsc{Challenge}$ procedure, hence $\Pr[\mathtt{Open}_s(i)] = \Pr[\mathtt{Open}_t(i)]$ for all $s, t \in [0, n]$, for all $i \in [n]$. Additionally, two consecutive hybrids $\mathsf{H}_k, \mathsf{H}_{k+1}$ only differ in the $k{+}1^{th}$ message returned during $\textsc{Challenge}$ *unless* $\mathcal{A}_{\mathsf{SO}}$ calls $\textsc{Open}(k{+}1)$ in game H_{k+1}. Thus, we have $\Pr[\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}} \Rightarrow 1 \wedge \mathtt{Open}_k(k+1)] = \Pr[\mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}} \Rightarrow 1 \wedge \mathtt{Open}_{k+1}(k+1)]$ and obtain

$$\mathbf{Adv}(\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}}) = \left| \Pr[\mathsf{H}_{k+1}^{\mathcal{A}_{\mathsf{SO}}} \Rightarrow 1 \wedge \overline{\mathtt{Open}_{k+1}(k+1)}] - \Pr[\mathsf{H}_k^{\mathcal{A}_{\mathsf{SO}}} \Rightarrow 1 \wedge \overline{\mathtt{Open}_k(k+1)}] \right|. \qquad (2)$$

We describe $\mathcal{B}_{\mathsf{mult}}$ (cf. Figure 7). $\mathcal{B}_{\mathsf{mult}}$ passes pk on to $\mathcal{A}_{\mathsf{SO}}$. Obtaining $(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})$, $\mathcal{B}_{\mathsf{mult}}$ makes a guess for middle – labeled middle^* – by making a guess – labeled N^* – of middle's neighbourhood in $G_{[k+1]}$ and samples $\mathbf{m}^0 \leftarrow \mathfrak{D}$. $\mathcal{B}_{\mathsf{mult}}$ resamples $\mathbf{m}^{1,0}$ fixing $N^* \cup \mathsf{right}$ and resamples $\mathbf{m}^{1,1}$ fixing $N^* \cup \mathsf{right} \cup \{k{+}1\}$. $\mathcal{B}_{\mathsf{mult}}$ sends $(\mathbf{m}^{1,0}_{\mathsf{middle}^*}, \mathbf{m}^{1,1}_{\mathsf{middle}^*})$ to its mult-IND-CPA challenger, receives $\mathbf{c}_{\mathsf{middle}^*}$, samples fresh randomness to encrypt messages in $\overline{\mathsf{middle}^*}$ on its own and forwards $(\mathbf{c}_1, \ldots, \mathbf{c}_n)$ to $\mathcal{A}_{\mathsf{SO}}$. $\mathcal{B}_{\mathsf{mult}}$ sets $\mathtt{Bad} := \mathit{true}$ if $\mathcal{A}_{\mathsf{SO}}$ calls $\textsc{Open}(i)$ for some $i \in \mathsf{middle}^* \setminus \{k+1\}$ since it cannot answer those queries.[2] Other opening queries are answered honestly. On $\mathcal{A}_{\mathsf{SO}}$'s call of $\textsc{Challenge}$, $\mathcal{B}_{\mathsf{mult}}$ checks if $N^* \subseteq \mathcal{I}$. If not, $\mathcal{B}_{\mathsf{mult}}$ guessed middle wrong and sets $\mathtt{Bad}$ to *true*. Otherwise, $\mathcal{B}_{\mathsf{mult}}$ resamples messages fixing those at positions $\mathcal{I} \cup \mathsf{right}$ to obtain resampled messages $\mathbf{m}^1$ and sends $\mathbf{m}^1_i$ for all left positions and $\mathbf{m}^0_i$ for all remaining positions to $\mathcal{A}_{\mathsf{SO}}$. $\mathcal{B}_{\mathsf{mult}}$ outputs whatever $\mathcal{A}_{\mathsf{SO}}$ outputs.

Assume that $\mathcal{B}_{\mathsf{mult}}$ guessed correctly, i.e. N^* is the neighbourhood of middle in $G_{[k]}$. Then $\mathsf{middle}^* = \mathsf{middle}$ holds and by definition of middle, $\mathtt{Bad}$ cannot happen.

Clearly, $\mathcal{B}_{\mathsf{mult}}$ correctly simulates $\mathcal{A}_{\mathsf{SO}}$'s hybrid view in all left and right positions. Note that $\mathcal{A}_{\mathsf{SO}}$ obtains *resampled* encryptions $\mathsf{Enc}_{pk}(\mathbf{m}^{1,b}_{\mathsf{middle}})$ during $\textsc{Enc}$, but expects *sampled* encryptions $\mathsf{Enc}_{pk}(\mathbf{m}^0_{\mathsf{middle}})$, while receiving *sampled* $\mathbf{m}^0_{\mathsf{middle}}$ on call of $\textsc{Challenge}$, expecting *resampled*

[2] Equation (2) directly accounts for $\mathcal{A}_{\mathsf{SO}}$ calling $\textsc{Open}(k+1)$.

Procedure INITIALIZE	**Procedure** OPEN(i)
$pk \leftarrow \text{INITIALIZE}_{\textsf{mult-IND-CPA}}(1^\lambda)$	if $i \in \textsf{middle}^* \setminus \{k+1\}$
Return pk	$\quad$ $\texttt{Bad} := true$
	$\mathcal{I} := \mathcal{I} \cup \{i\}$
Procedure ENC$(\mathfrak{D}, \textsf{Resamp}_{\mathfrak{D}})$	Return $(\mathbf{m}_i^0, \mathbf{r}_i)$
if $k < n-1$	
$\quad N^* \leftarrow_\$ \{V' \subseteq [k] \colon \|V'\| \in [0, B(G)-1]\}$	**Procedure** CHALLENGE
else	if $N^* \not\subseteq \mathcal{I}$
$\quad N^* \leftarrow_\$ \{V' \subseteq [k] \colon \|V'\| \in [0, B(G)]\}$	$\quad$ $\texttt{Bad} := true$
Let $\textsf{middle}^*$ denote the connected component in $G_{[k+1] \setminus N^*}$ that contains $k+1$.	$\mathbf{m}^1 \leftarrow \textsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I} \cup \textsf{right})$
$\mathbf{m}^0 \leftarrow \mathfrak{D}$	$\mathbf{m}_i = \begin{cases} \mathbf{m}_i^1 & \text{for } i \in \textsf{left} \\ \mathbf{m}_i^0 & \text{else} \end{cases}$
$\mathbf{m}^{1,0} \leftarrow \textsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, N^* \cup \textsf{right})$	Return $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$
$\mathbf{m}^{1,1} \leftarrow \textsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, N^* \cup \textsf{right} \cup \{k+1\})$	
$\mathbf{c}_{\textsf{middle}^*} \leftarrow \text{CHALLENGE}_{\textsf{mult-IND-CPA}}(\mathbf{m}^{1,0}_{\textsf{middle}^*}, \mathbf{m}^{1,1}_{\textsf{middle}^*})$	**Procedure** FINALIZE(b')
$\mathbf{r} \leftarrow_\$ \mathcal{R}^n$	$\text{FINALIZE}_{\textsf{mult-IND-CPA}}(b')$
$\mathbf{c}_i = \begin{cases} \mathbf{c}_i & \text{for } i \in \textsf{middle}^* \\ \textsf{Enc}_{pk}(\mathbf{m}_i^0; \mathbf{r}_i) & \text{else} \end{cases}$	
Return $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n)$	

Figure 7: $\mathcal{A}_{\textsf{SO}}$'s game interface as provided by $\mathcal{B}_{\textsf{mult}}$ run in the mult-IND-CPA game. $\mathcal{B}_{\textsf{mult}}$ interpolates between hybrids $\textsf{H}_k$, $\textsf{H}_{k+1}$ for $k \in [0, n-1]$.

$\mathbf{m}_{\textsf{middle}}$. Thus, *sampled* middle messages become *resampled* middle messages from $\mathcal{A}_{\textsf{SO}}$'s view and vice versa.

However, we have $\mathbf{m}_{\textsf{middle}} \equiv \mathbf{m}^0_{\textsf{middle}}$ since $N(\textsf{middle}) \subseteq \mathcal{I} \cup \textsf{right}$, whereby $\mathcal{I} \cup \textsf{right}$ is fixed when resampling $\mathbf{m}_{\textsf{middle}}$.

For $\mathcal{B}_{\textsf{mult}}$ run in the $\textsf{mult-IND-CPA}_{\textsf{PKE},0}$ game, $\mathcal{A}_{\textsf{SO}}$ receives $\textsf{Enc}_{pk}(\mathbf{m}^{1,0}_{\textsf{middle}})$ where $\mathbf{m}^{1,0}_{\textsf{middle}} \equiv \mathbf{m}^0_{\textsf{middle}}$ since $N^* \cup \textsf{right} = N \cup \textsf{right}$ is fixed when $\mathbf{m}^{1,0}$ is resampled. Hence, *all* middle messages sent during CHALLENGE look resampled and $\mathcal{A}_{\textsf{SO}}$'s view is identical to hybrid $\textsf{H}_{k+1}$.

When $\mathcal{B}_{\textsf{mult}}$ is run in $\textsf{mult-IND-CPA}_{\textsf{PKE},1}$, it forwards $\textsf{Enc}_{pk}(\mathbf{m}^{1,1}_{\textsf{middle}})$ to $\mathcal{A}_{\textsf{SO}}$ where $\mathbf{m}^{1,1}_{\textsf{middle}} \equiv \mathbf{m}^1_{\textsf{middle}}$ for the same reason as for $b = 0$. Especially, we have $\mathbf{m}^0_{k+1} = \mathbf{m}^{1,1}_{k+1}$ since $\mathbf{m}^0_{k+1}$ is fixed while resampling. Consequently, each messages in middle *except* the $k+1^{th}$ looks resampled during CHALLENGE and $\mathcal{A}_{\textsf{SO}}$'s view is identical to hybrid $\textsf{H}_k$. We have

$$\mathbf{Adv}^{\textsf{mult-IND-CPA}}_{\textsf{PKE}}(\mathcal{B}_{\textsf{mult}}, \lambda) \geq \left| \Pr[\textsf{mult-IND-CPA}^{\mathcal{B}_{\textsf{mult}}}_{\textsf{PKE},0} \Rightarrow 1] - \Pr[\textsf{mult-IND-CPA}^{\mathcal{B}_{\textsf{mult}}}_{\textsf{PKE},1} \Rightarrow 1] \right| .$$

$\mathcal{B}_{\textsf{mult}}$ outputs 1 in its mult-IND-CPA game if $\mathcal{A}_{\textsf{SO}}$ outputs 1 in its respective hybrid and $\mathcal{A}_{\textsf{SO}}$ does not open ciphertext $\mathbf{c}_{k+1}$ and $\texttt{Bad}$ does not happen:

$$= \left| \Pr[\textsf{H}^{\mathcal{A}_{\textsf{SO}}}_{k+1} \Rightarrow 1 \land \overline{\texttt{Open}_{k+1}(k{+}1)} \land \overline{\texttt{Bad}}] - \Pr[\textsf{H}^{\mathcal{A}_{\textsf{SO}}}_{k} \Rightarrow 1 \land \overline{\texttt{Open}_{k}(k{+}1)} \land \overline{\texttt{Bad}}] \right| .$$

Since $\overline{\texttt{Bad}}$ is independent of $\textsf{H}^{\mathcal{A}_{\textsf{SO}}}_{\kappa} \Rightarrow 1 \land \texttt{Open}_\kappa(k+1)$ for $\kappa \in \{k, k+1\}$ we have

$$= \Pr[\overline{\texttt{Bad}}] \cdot \left| \Pr[\textsf{H}^{\mathcal{A}_{\textsf{SO}}}_{k+1} \Rightarrow 1 \land \overline{\texttt{Open}_{k+1}(k+1)}] - \Pr[\textsf{H}^{\mathcal{A}_{\textsf{SO}}}_{k} \Rightarrow 1 \land \overline{\texttt{Open}_{k}(k+1)}] \right| .$$

$\mathcal{B}_{\mathsf{mult}}$ picks N^* from a set of size $\sum_{i=0}^{B(G\lambda)-1}\binom{k}{i}$ for $k < n-1$, of size $\sum_{i=0}^{B(G\lambda)}\binom{k}{i}$ for $k = n-1$, respectively and the claim of Lemma 3.12 follows by rearranging. ■

The remaining proof constitutes of tedious computations. We have

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \sum_{k=0}^{n-1} \Pr[\overline{\mathsf{Bad}}_k]^{-1} \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{mult\text{-}IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda).$$

Let $B := B(G)$. Since $\mathcal{B}_{\mathsf{mult}}$ submits message vectors of length $|\mathsf{middle}^*| \leq k+1$ to its $\mathsf{mult\text{-}IND\text{-}CPA}$ challenger and using Lemma 2.2:

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \left(\sum_{k=0}^{n-2}(k+1) \cdot \sum_{i=0}^{B-1}\binom{k}{i} + n \cdot \sum_{i=0}^{B}\binom{n-1}{i}\right) \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{mult}}, \lambda). \quad (3)$$

We upper-bound the loss in (3)

$$\sum_{k=0}^{n-2}(k+1) \cdot \sum_{i=0}^{B-1}\binom{k}{i} + n \cdot \sum_{i=0}^{B}\binom{n-1}{i} \leq 5 + \sum_{k=2}^{n-1}\underbrace{(k+1) \cdot \sum_{i=0}^{B-1} k^i}_{\leq 3\cdot(k^B-1)} + \sum_{i=0}^{B} n^{i+1}$$

$$\leq 8 - 3n + 3 \cdot \sum_{k=0}^{n-1} k^B + \sum_{i=0}^{B} n^{i+1} = 8 - 3n + 3 \cdot \sum_{k=0}^{n-1} k^B + \frac{n}{n-1} \cdot \left(n^{B+1} - 1\right) \qquad \text{for all } 2 \leq n.$$

$$\leq 6 - 3n + 3 \cdot \int_0^n k^B \mathrm{d}k + 2 \cdot n^{B+1} \leq 6 - 3n + \left(2 + \frac{3}{B+1}\right) \cdot n^{B+1} \leq 3 \cdot n^{B+1} \qquad \text{for all } 2 \leq B.$$

Since G is connected we have $B = 0 \Leftrightarrow n = 1$ and $B = 1 \Leftrightarrow n = 2$. Thus, it remains to verify and is easy to see that the bound holds for $(B, n) \in \{(0,1), (1,2)\}$ as well. ■

Because Markov distributions are DAG-induced by chain graphs and the maximum border of a chain graph is at most 2 we immediately obtain a tighter version of Corollary 3.6 whose proof directly follows from Theorem 3.11.

Corollary 3.13 *Let* PKE *be an* $\mathsf{IND\text{-}CPA}$ *secure public key encryption scheme, then* PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure with respect to efficiently resamplable Markov distributions over* $\mathcal{M}^n$.

In particular, for any adversary $\mathcal{A}_{\mathsf{SO}}$ *run in the* $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE}}$ *game, there exists an* $\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$ *adversary* $\mathcal{B}_{\mathsf{CPA}}$ *with roughly the same running time as* $\mathcal{A}_{\mathsf{SO}}$ *such that*

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq 3 \cdot n^3 \cdot \mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{CPA}}, \lambda).$$

Applying the same proof of 3.11 directly to the Markov case gives a slightly better bound on the loss, namely: $n \cdot (n+1) \cdot (2n+1)/6$, since $N(\mathsf{middle}) \cap [n-1] = 1$ even for the last transition H_{n-1} to H_n. Hence, the loss in Equation (3) boils down to $\sum_{k=0}^{n-1}(k+1)^2$.

Recall that the hybrids in the proof of Theorem 3.11 saved us a factor of n because it suffices to guess a set of size at most $B(G) - 1$ instead of $B(G)$ for $k < n-1$ as at least one vertex of middle's neighbourhood is contained in right.

Procedure INITIALIZE $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ Return pk	**Procedure** OPEN(i) $\mathcal{I} := \mathcal{I} \cup \{i\}$ Return $(\mathbf{m}_i^0, \mathbf{r}_i)$
Procedure ENC$(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})$ $\mathbf{m}^0 \leftarrow \mathfrak{D}$ $\mathbf{r} \leftarrow_\$ \mathcal{R}^n$ $\mathbf{c} = \mathsf{Enc}_{pk}(\mathbf{m}^0_{V_z}; \mathbf{r}_{V_z})$ Return $\mathbf{c}$	**Procedure** CHALLENGE $\mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I})$ Return $\mathbf{m}^b_{V_z}$ **Procedure** FINALIZE(b') Return b'

Figure 8: $\mathcal{B}_{\mathsf{G\text{-}SO}}$'s interface in the $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},b,z}$ game.

The same hybrids can be used to strengthen Theorem 3.3 as it suffices to guess a connected subgraph in $[k+1]$ (instead of $[n]$) containing vertex $k+1$.

Since G is connected, there is at least a path in $\{k+1\} \cup \mathsf{right}$ that contains $k+1$, i.e. at least $n-k$ connected subgraphs in $\mathsf{right} \cup \{k+1\}$. Thus, there exist at least $n-k$ connected subgraphs in G that contain vertex $k+1$ and are identical if restricted to $[k+1]$. Hence the reduction's probability of guessing C_{k+1} correctly can be increased from $1/S(G)$ to $(n-k)/S(G)$, merely bringing the loss from $\mathcal{O}(n^2) \cdot S(G)$ down to $\mathcal{O}(n \cdot \log n) \cdot S(G)$.

3.6 A Hybrid Argument for Disconnected Graphs

For a graph G on z' connected components fix any semi-order on them, e.g. ordered by the smallest vertex in each component and let $V_1, \ldots, V_{z'}$ denote the vertices of the connected components of G. For $j \in [z'+1, n]$ let $V_j := \emptyset$. We define a security game where an adversary plays the $\mathsf{IND\text{-}SO\text{-}CPA}$ game on a connected component of the graph that induced the distribution chosen by the adversary.

Definition 3.14 For a public-key encryption scheme $\mathsf{PKE} := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, a bit b, a family $\mathcal{F}$ of efficiently resamplable, G-induced distributions over $\mathcal{M}^n$, $z \in [n]$ and an adversary $\mathcal{B}_{\mathsf{G\text{-}SO}}$ we consider the $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}^{\mathcal{B}_{\mathsf{G\text{-}SO}}}_{\mathsf{PKE},b,z}$ game given in Figure 8. Run in the game, $\mathcal{B}_{\mathsf{G\text{-}SO}}$ calls ENC once right after INITIALIZE and submits $\mathfrak{D} \in \mathcal{F}$ along with a PPT resampling algorithm $\mathsf{Resamp}_{\mathfrak{D}}$. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ may call OPEN multiple times but only for $i \in V_z$ and invokes CHALLENGE once after its last OPEN query before calling FINALIZE. We define the advantage of $\mathcal{B}_{\mathsf{G\text{-}SO}}$ run in the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},b,z}$ game as

$$\mathbf{Adv}^{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE},z}(\mathcal{B}_{\mathsf{G\text{-}SO}}, \mathfrak{D}_\lambda, \lambda) := \mathbf{Adv}(\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}^{\mathcal{B}_{\mathsf{G\text{-}SO}}}_{\mathsf{PKE},0,z}, \mathsf{G\text{-}IND\text{-}SO\text{-}CPA}^{\mathcal{B}_{\mathsf{G\text{-}SO}}}_{\mathsf{PKE},1,z}).$$

PKE *is* $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_z$ *secure w.r.t.* $\mathcal{F}$ if $\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE},z}(\mathcal{B}_{\mathsf{G\text{-}SO}}, \mathfrak{D}_\lambda, \lambda)$ is negligible for all PPT adversaries $\mathcal{B}_{\mathsf{G\text{-}SO}}$. PKE *is* $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}$ *secure w.r.t.* $\mathcal{F}$ if PKE is $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_z$ secure w.r.t. $\mathcal{F}$ for all $z \in [n]$.

We have $\mathbf{Adv}^{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE},z}(\mathcal{B}_{\mathsf{G\text{-}SO}}, \mathfrak{D}_\lambda, \lambda) = 0$ for $z \in [z'+1, n]$.

Theorem 3.15 *Let* PKE *be* $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}$ *secure w.r.t. a family* $\mathcal{F}$ *of efficiently resamplable and* G*-induced distributions over* $\mathcal{M}^n$*, then* PKE *is* $\mathsf{IND\text{-}SO\text{-}CPA}$ *secure w.r.t* $\mathcal{F}$*.*

Proof of Theorem 3.15: Again, the main idea is that connected components can be dealt with independently. We give a hybrid argument over the connected components of G_λ using $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_z$ security for switching connected component z from sampled to resampled. See Figure 9 for the hybrid's code of Challenge, every other procedure stays as in the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},b}$ game (cf. Figure 2).

$$\begin{array}{|l|}\hline \underline{\textbf{Procedure } \textsc{Challenge}} \\ \mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I}) \\ \mathbf{m}_i = \begin{cases} \mathbf{m}_i^1 & \text{for } i \in \bigcup_{j=1}^{z} V_j \\ \mathbf{m}_i^0 & \text{else} \end{cases} \\ \text{Return } \mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_n) \\ \hline \end{array}$$

Figure 9: Hybrid H_z. The first z connected components are already resampled conditioned on opening queries, while the remaining are still sampled.

Note that H_0 (resp. $\mathsf{H}_{z'}$) is identical to the $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},0}$ (resp. $\mathsf{IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},1}$) game. Thus

$$\mathbf{Adv}^{\mathsf{IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) = \mathbf{Adv}(\mathsf{H}_0^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{z'}^{\mathcal{A}_{\mathsf{SO}}}) \le \sum_{z=0}^{z'-1} \mathbf{Adv}(\mathsf{H}_z^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{z+1}^{\mathcal{A}_{\mathsf{SO}}}).$$

We proceed with the following Lemma.

Lemma 3.16 *For every adversary $\mathcal{A}_{\mathsf{SO}}$ distinguishing hybrids H_z and H_{z+1} there exists an adversary $\mathcal{B}_{\mathsf{G\text{-}SO}}$ run in the $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},z+1}$ with roughly the same running time such that*

$$\mathbf{Adv}(\mathsf{H}_z^{\mathcal{A}_{\mathsf{SO}}}, \mathsf{H}_{z+1}^{\mathcal{A}_{\mathsf{SO}}}) \le \mathbf{Adv}^{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}}_{\mathsf{PKE},z+1}(\mathcal{B}_{\mathsf{G\text{-}SO}}, \mathfrak{D}_\lambda, \lambda).$$

Proof of Lemma 3.16: We construct an adversary $\mathcal{B}_{\mathsf{G\text{-}SO}}$ that interpolates between hybrids H_z and H_{z+1} for $\mathcal{A}_{\mathsf{SO}}$. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ proceeds as follows (cf. Figure 10).
$\mathcal{B}_{\mathsf{G\text{-}SO}}$ forwards pk to $\mathcal{A}_{\mathsf{SO}}$. On $\mathcal{A}_{\mathsf{SO}}$'s call of Enc, $\mathcal{B}_{\mathsf{G\text{-}SO}}$ calls $\textsc{Enc}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}$ to obtain an encryption $\mathbf{c}_{V_{z+1}}$ of messages in the V_{z+1}. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ samples messages $\mathbf{m}^0 \leftarrow \mathfrak{D}$ on its own and encrypts the messages in $\overline{V_{z+1}}$. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ sends $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n)$ to $\mathcal{A}_{\mathsf{SO}}$. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ answers opening queries on its own unless they occur on V_{z+1}, where it invokes its $\textsc{Open}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}$ oracle to answer. On Challenge, $\mathcal{B}_{\mathsf{G\text{-}SO}}$ receives a challenge message vector $\mathbf{m}_{V\,z+1}$ by calling $\textsc{Challenge}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}$ and resamples $\mathbf{m}^0$ conditioned on $\mathcal{I}$. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ returns resampled messages $\mathbf{m}^1$ on $\cup_{j=1}^{z} V_j$, its challenge messages $\mathbf{m}_{V\,z+1}$ and sampled messages $\mathbf{m}^0$ for $\cup_{j=z+2}^{n} V_j$ to $\mathcal{A}_{\mathsf{SO}}$. $\mathcal{B}_{\mathsf{G\text{-}SO}}$ outputs whatever $\mathcal{A}_{\mathsf{SO}}$ outputs.

Obviously $\mathcal{B}_{\mathsf{G\text{-}SO}}$ simulates the hybrids correctly during Enc since it always returns encryptions of sampled messages. On $\mathcal{A}_{\mathsf{SO}}$'s call of Challenge the messages in the first z connected components are already resampled while the messages in the last $n - z - 1$ connected components are sampled as in hybrids H_z and H_{z+1}. If $\mathcal{B}_{\mathsf{G\text{-}SO}}$ is run in the $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},0,z+1}$ game, it obtains sampled messages for the $z + 1^{th}$ connected component, thus it runs $\mathcal{A}_{\mathsf{SO}}$ in hybrid H_z while $\mathcal{B}_{\mathsf{G\text{-}SO}}$ receives resampled messages for V_{z+1} when run in $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},1,z+1}$, hence running $\mathcal{A}_{\mathsf{SO}}$ in hybrid H_{z+1}. Thus

$$\Pr[\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}^{\mathcal{B}_{\mathsf{G\text{-}SO}}}_{\mathsf{PKE},0,z+1} \Rightarrow 1] = \Pr[\mathsf{H}_z^{\mathcal{A}_{\mathsf{SO}}} \Rightarrow 1]$$

Procedure $\underline{\text{INITIALIZE}}$ $pk \leftarrow \mathsf{Gen}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}(1^\lambda)$ Return pk **Procedure** $\underline{\text{ENC}(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})}$ $\mathbf{c}_{V_{z+1}} \leftarrow \text{ENC}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}(\mathfrak{D}, \mathsf{Resamp}_{\mathfrak{D}})$ $\mathbf{m}^0 \leftarrow \mathfrak{D}$ $\mathbf{r} \leftarrow_\$ \mathcal{R}^n$ $\mathbf{c}_i = \begin{cases} \mathbf{c}_i & \text{for } i \in V_{z+1} \\ \mathsf{Enc}_{pk}(\mathbf{m}_i^0; \mathbf{r}_i) & \text{else} \end{cases}$ Return $\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_n)$ **Procedure** $\underline{\text{FINALIZE}(b')}$ $\text{FINALIZE}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}}(b')$	**Procedure** $\underline{\text{OPEN}(i)}$ $\mathcal{I} := \mathcal{I} \cup \{i\}$ if $i \in V_{z+1}$ Return $\text{OPEN}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}(i)$ else Return $(\mathbf{m}_i^0, \mathbf{r}_i)$ **Procedure** $\underline{\text{CHALLENGE}}$ $\mathbf{m}_{V_{z+1}} \leftarrow \text{CHALLENGE}_{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{z+1}}$ $\mathbf{m}^1 \leftarrow \mathsf{Resamp}_{\mathfrak{D}}(\mathbf{m}^0, \mathcal{I})$ $\mathbf{m}_i = \begin{cases} \mathbf{m}_i^1 & \text{for } i \in \cup_{j=1}^{z} V_j \\ \mathbf{m}_i & \text{for } i \in V_{z+1} \\ \mathbf{m}_i^0 & \text{else} \end{cases}$ Return $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$

Figure 10: Reduction run by $\mathcal{B}_{\mathsf{G\text{-}SO}}$ to simulate H_z (resp. H_{z+1}) when $\mathcal{B}_{\mathsf{G\text{-}SO}}$ is run in $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},0,z+1}$ (resp. $\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},1,z+1}$).

and

$$\Pr[\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}_{\mathsf{PKE},1,z+1}^{\mathcal{B}_{\mathsf{G\text{-}SO}}} \Rightarrow 1] = \Pr[\mathsf{H}_{z+1}^{\mathcal{A}_{\mathsf{SO}}} \Rightarrow 1].$$

Lemma 3.16 follows. ■

We obtain

$$\mathbf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}SO\text{-}CPA}}(\mathcal{A}_{\mathsf{SO}}, \mathfrak{D}_\lambda, \lambda) \leq \sum_{z=1}^{z'} \mathbf{Adv}_{\mathsf{PKE},z}^{\mathsf{G\text{-}IND\text{-}SO\text{-}CPA}}(\mathcal{B}_{\mathsf{G\text{-}SO}}, \mathfrak{D}_\lambda, \lambda)$$

and Theorem 3.15 follows immediately since $z' \leq n$. ■

In particular we achieve versions of Theorem 3.3, Corollary 3.10 and Theorem 3.11 for disconnected graphs, whereby

$$S(G) = \sum_{i=1}^{z'} S(C_i) \quad \text{and} \quad B(G) = \max_{i \in [z']}\{B(C_i)\}$$

for a graph G consisting of connected components $C_1, \ldots, C_{z'}$.

Moreover, for $G = ([n], \emptyset)$, G-induced distributions become product distributions, i.e. the messages are sampled independently. Hence, the positive result of [3] can be seen as a special case of Theorem 3.15.

References

[1] M. Bellare, R. Dowsley, B. Waters, and S. Yilek. Standard security does not imply security against selective-opening. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 645–662, Cambridge, UK, Apr. 15–19, 2012. Springer, Berlin, Germany. (Cited on page 3.)

[2] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In A. Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35, Cologne, Germany, Apr. 26–30, 2009. Springer, Berlin, Germany. (Cited on page 3, 6.)

[3] M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. *IACR Cryptology ePrint Archive*, 2009:101, 2009. (Cited on page 2, 3, 19.)

[4] F. Böhl, D. Hofheinz, and D. Kraschewski. On definitions of selective opening security. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 522–539, Darmstadt, Germany, May 21–23, 2012. Springer, Berlin, Germany. (Cited on page 2, 3, 6.)

[5] J. M. Borwein and O.-Y. Chan. Uniform bounds for the complementary incomplete gamma function. (Cited on page 12.)

[6] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable encryption. In B. S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 90–104, Santa Barbara, CA, USA, Aug. 17–21, 1997. Springer, Berlin, Germany. (Cited on page 3.)

[7] R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648, Philadephia, Pennsylvania, USA, May 22–24, 1996. ACM Press. (Cited on page 3.)

[8] C. Dwork, M. Naor, O. Reingold, and L. J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534, New York, New York, USA, Oct. 17–19, 1999. IEEE Computer Society Press. (Cited on page 2, 3.)

[9] D. Hofheinz, V. Rao, and D. Wichs. Standard security does not imply indistinguishability under selective opening. Cryptology ePrint Archive, Report 2015/792, 2015. `http://eprint.iacr.org/`. (Cited on page 3.)

[10] D. Hofheinz and A. Rupp. Standard versus selective opening security: Separation and equivalence results. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 591–615, San Diego, CA, USA, Feb. 24–26, 2014. Springer, Berlin, Germany. (Cited on page 3.)

[11] P. Natalini and B. Palumbo. Inequalities for the incomplete gamma function. *Math. Inequal. Appl.*, 3:69–77, 2000. (Cited on page 12.)