A preliminary version of this paper appears in the proceedings of FSE 2015. This is the full version.

A More Cautious Approach to Security Against Mass Surveillance

Jean Paul Degabriele[1], Pooya Farshim[2], and Bertram Poettering[3]

[1] Royal Holloway, University of London, United Kingdom
[2] Queen's University Belfast, United Kingdom
[3] Ruhr University Bochum, Germany

Abstract. At CRYPTO 2014 Bellare, Paterson, and Rogaway (BPR) presented a formal treatment of symmetric encryption in the light of algorithm substitution attacks (ASAs), which may be employed by 'big brother' entities for the scope of mass surveillance. Roughly speaking, in ASAs big brother may bias ciphertexts to establish a covert channel to leak vital cryptographic information. In this work, we identify a seemingly benign assumption implicit in BPR's treatment and argue that it artificially (and severely) limits big brother's capabilities. We then demonstrate the critical role that this assumption plays by showing that even a slight weakening of it renders the security notion completely unsatisfiable by *any*, possibly deterministic and/or stateful, symmetric encryption scheme. We propose a refined security model to address this shortcoming, and use it to restore the positive result of BPR, but caution that this defense does not stop most other forms of covert-channel attacks.

Keywords. Mass surveillance, algorithm substitution attack, symmetric encryption, covert channel.

1 Introduction

In 2013 Edward Snowden shocked the world with revelations of several ongoing surveillance programs targeting citizens worldwide [1,9]. There is now incontestable evidence that national intelligence agencies can go to great lengths to undermine our privacy. The methods employed to attack and infiltrate our communication infrastructure are rather disturbing. Amongst others these include sabotaging Internet routers, wire-tapping international undersea cables, installing backdoors in management front ends of telecom providers, injecting malware in real-time into network packets carrying executable files, and intercepting postal shipping to replace networking hardware.

Some of the revelations concern the domain of cryptography. Somewhat reassuringly, there was no indication that any of the well-established cryptographic primitives and hardness assumptions could be broken by the national intelligence agencies. Instead these agencies resorted to more devious means in order to compromise the security of cryptographic protocols. In one particular instance the National Security Agency (NSA) manoeuvred cryptographic standardization bodies to recommend a cryptographic primitive which contained a backdoor [12]:

The specification of the `Dual_EC_DRBG` cryptographic random-number generator [2] contains arbitrary-looking parameters for which there exists trapdoor information, known to its creators, that can be used to predict future results from a sufficiently long stretch of output [15]. A recent study [5] explores the practicality of exploiting this vulnerability in TLS. In particular it shows that support of the Extended Random TLS extension [13] (an IETF draft co-authored by an NSA employee) makes the vulnerability much easier to exploit. Furthermore the NSA is known to have made secret payments to vendors in order to include the `Dual_EC_DRBG` in their products and increase proliferation [10].

Such tactics clearly fall outside of the threat models that we normally assume in cryptography and call for a reconsideration of our most basic assumptions. It is hence natural to ask what other means could be employed by such powerful entities to subvert cryptographic protocols. Recent work by Bellare, Paterson and Rogaway [4] explores the possibility of mass surveillance through *algorithm substitution attacks* (ASA). Consider some type of closed-source software making use of a standard symmetric encryption scheme to achieve its security goals. In an ASA the standard encryption scheme is substituted with an alternative scheme that the attacker has authored; we call this latter scheme a *subversion*. A successful ASA would allow the adversary, henceforth referred to as *big brother*, to undermine the confidentiality of the data and at the same time circumvent *detection* by its users.

BPR's treatment. Bellare, Paterson and Rogaway (BPR) [4] define a formal framework for analyzing resistance to a certain class of ASAs in the context of symmetric encryption. At a very high level, their notion of surveillance resistance requires that big brother be incapable of distinguishing ciphertext produced by the legitimate scheme to ciphertexts produced by the subverted scheme. They also put forward a notion of undetectability, that can be seen as a dual of the former notion, which guarantees that no efficient detection algorithm is capable of distinguishing legitimate ciphertext from those produced by the subverted scheme. This latter notion is only intended to prove *negative* results. That is, we essentially have no hope of resisting ASAs that meet this notion undetectability. BPR are able to establish a set of positive and negative results within their framework. They build on the work of [8] to demonstrate ASAs on specific schemes, such as the modes of operation CTR$ and CBC$. Their negative results culminate with the *biased-ciphertext attack* which can be mounted against any randomized symmetric encryption scheme that uses a sufficient amount of randomness. This attack establishes a *covert channel* between the subverted encryption algorithm and big brother, through which he is able to retrieve the full user key. Furthermore the biased-ciphertext attack is shown to be undetectable, inidicating that no probabilistic encryption scheme can resist ASAs. Accordingly, BPR turn to stateful deterministic schemes and identify a combinatorial property that is sufficient to ensure security within their security model. Most modern nonce-based schemes [14] can be easily shown to satisfy this property.

Contributions. In this work we revisit the security model proposed by BPR [4] and re-examine its underlying assumptions. Our main criticism concerns the notion of *perfect decryptability*, and the requirement that every *subversion* must satisfy it. Decryptability is introduced as a minimal requirement that a subversion must meet in order to have some chance of avoiding detection. Accordingly, the assumption is that big brother would only consider subversions that satisfy this condition. We argue, however, that this requirement is stronger than what is substantiated by this rationale, and it results in artificially limiting big brother's set of available strategies. Indeed, we show that with a minimal relaxation of the decryptability condition the BPR security notion becomes totally unsatisfiable. More precisely, for *any* symmetric encryption scheme, deterministic or not, we construct a corresponding undetectable subversion that can be triggered to leak information when run on specific inputs known solely by big brother. From a theoretical perspective this shows that the instantiability of the security model crucially depends on this requirement. From a more practical perspective, security in the BPR model simply does not translate to security in practice.

As pointed out in [4], defending against ASAs requires the ability to detect them. Indeed, the ability to detect an ASA is an important measure of security which should be surfaced by the security definition. We observe that in this respect the BPR security definition falls short – encryption schemes are considered secure as long as subversions can be detected with *non-zero* probability. A scheme guaranteeing a detection probability of 2^{-128}, say, is of little practical significance but in the BPR model it is deemed secure.

Building on the work of Bellare, Paterson and Rogaway [4] we propose an alternative security definition to address the above shortcomings. Our model disposes of the perfect decryptability requirement and instead quantifies security via a new detectability notion. In particular we require that a scheme come with a corresponding detection algorithm $\mathscr{U}$. The detection game starts by running big brother as in the BPR surveillance game and a transcript of his queries is maintained. The detection algorithm is then given the user key and the transcript and will determine whether a subversion has taken place.

Although in our security definition the detector runs after big brother, the implemented detection strategy need not necessarily be after the fact. The detector could be run each time a new ciphertext is computed and only transmitted if no anomaly is detected. Such a 'live' detection strategy appears to be necessary since one-time detection strategies (as proposed in [4]) are not effective against the input-triggered subversions discussed above. We define security by requiring that for any subversion the detector's advantage must be quantitavely comparable to big brother's surveillance advantage. We then re-establish the relative strength of deterministic stateful schemes in comparison to randomized ones, suggested in [4], in our security model.

Limitations of the security model. Our main goal here is to point out and address shortcomings in the security model proposed in [4]. Accordingly, we try to deviate as least as possible from the setting considered therein and its underlying assumptions. Bellare, Paterson and Rogaway state very clearly the

restricted scope of their analysis, and naturally these restrictions are inhereted in our analysis as well. In particular, information can only be leaked to big brother through the ciphertexts, and other forms of covert channels, based on ciphertext timings or power consumption, are not addressed in our analysis. Similarly we only look at symmetric encryption, whereas real-world security protocols are constructed from more cryptographic components. Thus protecting symmetric encryption from subversion does not gurantee that the security protocols in which they are used are protected against subversion. Nonetheless, we believe that [4] and our work are important first steps towards better understanding the problems relating to mass surveillance and the limitations in protecting against it.

Other related work. The first systematic analysis of how malicious modification of implemented cryptosystems can weaken their expected security dates back to Simmons [16]. He studied how cryptographic algorithms in black-box implementations can be made to leak information about secret keying material via *subliminal channels*. In the setting considered by Simmons, anyone who successfully reverse-engineers the manipulated code would also be able to recover the leaked secrets.

This was refined by Young and Yung in a sequence of works [17,18,19,20,21,22] under the theme of *Kleptography*, covering mainly primitives in the realm of public-key cryptography (encryption and signature schemes based on RSA and DLP). In their proposals for protocol subversion, a central part of the injected algorithms is the public key of the attacker to which all leakage is 'safely encrypted'. The claim is then that if the implementation is successfully reverse-engineered reveals the existence of a backdoor, the security of the overall system does not collapse, as the attacker's secret key would be held responsibly (by, say, a governmental agency). Kleptographic attacks on RSA systems were also reported by Crépeau and Slakmon [6] who optimized the efficiency of subverted key-generation algorithms by using symmetric techniques. Concerning higher-level protocols, algorithm substitution attacks targeting specifically the SSL/TLS and SSH protocols were reported by Goh et al. [8], and Young and Yung [23].

2 Preliminaries.

Notation. Unless otherwise stated, an algorithm may be randomized. An adversary is an algorithm. For any algorithm $\mathscr{A}$, $y \leftarrow \mathscr{A}(x_1, x_2, \dots)$ denotes executing $\mathscr{A}$ with fresh coins on inputs $x_1, x_2, \dots$ and assigning its output to y. For n, a positive integer, we use $\{0,1\}^n$ to denote the set of all binary strings of length n and $\{0,1\}^*$ to denote the set of all finite binary strings. The empty string is represented by ε. For any two strings x and y, $x \parallel y$ denotes their concatenation and $|x|$ denotes the length of x. For any vector $\mathbf{X}$, we denote by $\mathbf{X}[i]$ its i^{th} component. If $\mathcal{S}$ is a finite set then $|\mathcal{S}|$ denotes its size, and $y \leftarrow\!\!\$\ \mathcal{S}$ denotes the process of selecting an element from $\mathcal{S}$ uniformly at random and assigning it to y. $\Pr[\, P : E \,]$ denotes the probability of event E occurring after having

executed process P. Security definitions are formulated through the code-based game-playing framework.

SYMMETRIC ENCRYPTION. A *symmetric encryption scheme* is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Associated to Π are the message space $\mathcal{M} \subseteq \{0,1\}^*$ and the associated data space $\mathcal{AD} \subseteq \{0,1\}^*$. The *key space* $\mathcal{K}$ is a non-empty set of strings of some fixed length. The *encryption algorithm* $\mathcal{E}$ may be randomized, stateful, or both. It takes as input the secret key $K \in \mathcal{K}$, a message $M \in \{0,1\}^*$, an associated data $A \in \{0,1\}^*$, and the current encryption state σ to return a ciphertext C or the special symbol $\perp$, together with an updated state. The symbol $\perp$ may be returned for instance if $M \notin \mathcal{M}$ or $A \notin \mathcal{AD}$. The *decryption algorithm* $\mathcal{D}$ is deterministic but may be stateful. It takes as input the secret key K, a ciphertext string $C \in \{0,1\}^*$, an associated data string $A \in \{0,1\}^*$, and the current decryption state ϱ to return the corresponding message M or the special symbol $\perp$, and an updated state. Pairs of ciphertext and associated data that result in $\mathcal{D}$ outputting $\perp$ are called *invalid*.

The encryption and decryption states are always initialized to ε. For either of $\mathcal{E}$ or $\mathcal{D}$, we say that it is a stateless algorithm if for all inputs in $\mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \times \{\varepsilon\}$ the returned updated state is always ε. The scheme Π is said to be stateless if both $\mathcal{E}$ and $\mathcal{D}$ are stateless. We require that for any $M \in \mathcal{M}$ and any $A \in \mathcal{AD}$ it holds that $\{0,1\}^{|M|} \subseteq \mathcal{M}$ and $\{0,1\}^{|A|} \subseteq \mathcal{AD}$.

For any symmetric encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, any $\ell \in \mathbb{N}$, any vector $\mathbf{M} = [M_1, \ldots, M_\ell] \in \mathcal{M}^\ell$ and any vector $\mathbf{A} = [A_1, \ldots, A_\ell] \in \mathcal{AD}^\ell$, we write $(\mathbf{C}, \sigma_\ell) \leftarrow \mathcal{E}_K(\mathbf{M}, \mathbf{A}, \varepsilon)$ as shorthand for:

$$(C_1, \sigma_1) \leftarrow \mathcal{E}_K(M_1, A_1, \varepsilon);\ \ldots;\ (C_\ell, \sigma_\ell) \leftarrow \mathcal{E}_K(M_\ell, A_\ell, \sigma_{\ell-1})\ ,$$

where $\mathbf{C} = [C_1, \ldots, C_\ell]$. Similarly we write $(\mathbf{M}', \varrho_\ell) \leftarrow \mathcal{D}_K(\mathbf{C}, \mathbf{A}, \varepsilon)$ to denote the analogous process for decryption.

Definition 1 (Correctness [4]). *A symmetric encryption scheme* Π *is said to be* (q, δ)-correct *if for all* $\ell \leq q$*, all* $\mathbf{M} \in \mathcal{M}^\ell$ *and all* $\mathbf{A} \in \mathcal{AD}^\ell$*, it holds that:*

$$\Pr\left[\, K \xleftarrow{\$} \mathcal{K}; (\mathbf{C}, \sigma_\ell) \leftarrow \mathcal{E}_K(\mathbf{M}, \mathbf{A}, \varepsilon); (\mathbf{M}', \varrho_\ell) \leftarrow \mathcal{D}_K(\mathbf{C}, \mathbf{A}, \varepsilon) \,:\, \mathbf{M} \neq \mathbf{M}' \,\right] \leq \delta\,.$$

Schemes that achieve correctness with $\delta = 0$ *for all* $q \in \mathbb{N}$ *are said to be* perfectly correct*.*

We now recall the standard IND-CPA security notion for symmetric encryption [3].

Definition 2 (Privacy). *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a symmetric encryption scheme and let* $\mathscr{A}$ *be an adversary. Consider the game* $\text{IND-CPA}_\Pi^{\mathscr{A}}$ *depicted in Figure 1. The adversary's advantage is defined as*

$$\mathbf{Adv}_\Pi^{\text{ind-cpa}}(\mathscr{A}) := 2 \cdot \Pr\left[\, \text{IND-CPA}_\Pi^{\mathscr{A}} \,\right] - 1\,.$$

The scheme Π *is said to be* ϵ*-private if for every practical adversary* $\mathscr{A}$ *its advantage* $\mathbf{Adv}_\Pi^{\text{ind-cpa}}(\mathscr{A})$ *is bounded by* ϵ*.*

Intuitively, when ϵ is sufficiently small we may simply say that Π is IND-CPA secure.

Game IND-CPA$^{\mathscr{A}}_{\Pi}$	$\textsc{Enc}(M_0, M_1, A)$
$b \leftarrow\!\!\$\ \{0,1\}$	if $\lvert M_0\rvert \neq \lvert M_1\rvert$ then return $\perp$
$\sigma \leftarrow \varepsilon; K \leftarrow\!\!\$\ \mathcal{K}$	$(C, \sigma) \leftarrow \mathcal{E}(K, M_b, A, \sigma)$
$b' \leftarrow \mathscr{A}^{\textsc{Enc}}$	return C
return $(b = b')$	

Fig. 1: Game defining the IND-CPA security of scheme Π against $\mathscr{A}$.

3 Algorithm Substitution Attacks

In an algorithm substitution attack (ASA), big brother is able to covertly replace the code of an encryption algorithm $\mathcal{E}(K, \ldots)$ (forming part of some wider protocol) with the subverted encryption algorithm $\widetilde{\mathcal{E}}(\widetilde{K}, K, \ldots)$. Here, $\widetilde{\mathcal{E}}$ takes the same inputs as $\mathcal{E}$ together with a subversion key $\widetilde{K}$ which is assumed to be embedded in the code in an obfuscated manner, and hence is inaccessible to users. Intuitively, the subversion key significantly improves big brother's ability to leak information via the ciphertexts without being detected. For instance, it can use $\widetilde{K}$ to encrypt a user's key and use the result as a random-looking IV in the ciphertext. Big brother can later intercept this ciphertext, recover the user's key from the IV, and use it to decrypt the rest of the ciphertexts. In addition allow the operations of $\widetilde{\mathcal{E}}$ to depend on user-specific identification parameter i.

Note that when considering ASAs the concern is not about whether the real encryption scheme contains a backdoor, possibly due to an obscurely generated set of parameters. In fact an inherent assumption in the setting proposed in [4], and in this paper, is that the real encryption scheme $\mathcal{E}$ achieves the required level of security and in particular is free from backdoors. Instead, the question being asked is whether an *implementation* of the real scheme, possibly obfuscated, contains a backdoor and under what circumstances this can be detected.

Subversions. For any symmetric encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ its subversion is a pair $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}})$. The *subversion key space* $\widetilde{\mathcal{K}}$ is a finite non-empty set. The *subverted encryption algorithm* $\widetilde{\mathcal{E}}$ may be randomized, stateful, or both. It takes as input a subversion key $\widetilde{K} \in \widetilde{\mathcal{K}}$, a user's secret key $K \in \mathcal{K}$, a message $M \in \{0,1\}^*$, an associated data string $A \in \{0,1\}^*$, an encryption state σ, and a user identifier i, to return a ciphertext C or the special symbol $\perp$, together with an updated state. The user identifier is considered a public value such as an IP or MAC address unique to each key K; the precise value of i does not matter—it is merely used to label users.

In order for the subversion to be of any use to big brother, there must exist some associated *plaintext-recovery algorithm* which uses the subversion key to recover information from the ciphertexts. In [4] this algorithm was denoted by $\widetilde{\mathcal{D}}$. Its success in recovering the corresponding plaintexts may vary in a number of ways. From the users' standpoint we only concern ourselves to bound this success probability by a small quantity. This is dealt with in the security definitions where any such plaintext-recovery algorithm is implicit in the adversary.

3.1 The Bellare–Paterson–Rogaway Security Model

Bellare, Paterson and Rogaway (BPR) [4] formulate security against ASAs as follows. In addition to the standard security requirements for symmetric encryption they consider two new notions, namely *undetectability* and *surveillance* defined respectively using the DETECT and SURV games shown in Figure 2.

In DETECT an algorithm $\mathscr{U}$, representing a detection test, is required to distinguish $\mathcal{E}$ from $\widetilde{\mathcal{E}}_{\widetilde{K}}$ when given oracle access to one of these two algorithms. More specifically, a bit b and a subversion key $\widetilde{K}$ are first sampled and $\mathscr{U}$ is then given access to two oracles, KEY and ENC. The game models a multi-user setting and the KEY oracle serves to let user i create a secret key. The ENC oracle takes a message M, associated data A, and a user identifier i and depending on the value of b it returns an encryption under either $\mathcal{E}$ or $\widetilde{\mathcal{E}}_{\widetilde{K}}$. The game ends when $\mathscr{U}$ halts and outputs a bit b' as its guess of bit b. The corresponding advantage is defined as:

$$\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) := 2 \cdot \Pr\left[\, \mathrm{DETECT}^{\mathscr{U}}_{\Pi,\widetilde{\Pi}} \,\right] - 1\,.$$

In SURV an adversary $\mathscr{B}$, who does not have access to the users' secret keys but knows the subversion key, is required to distinguish $\mathcal{E}$ from $\widetilde{\mathcal{E}}_{\widetilde{K}}$ when given oracle access to one of these algorithms. The game proceeds by first sampling a bit b and a subversion key $\widetilde{K}$, and then $\mathscr{B}$ is given access to $\widetilde{K}$ and two oracles, KEY and ENC. Oracle KEY only serves to initialize a secret key for specified user i and does not return any value. The ENC oracle takes a message M, associated data A, and a user identifier i, and depending on the value of b it returns an encryption under either $\mathcal{E}$ or $\widetilde{\mathcal{E}}_{\widetilde{K}}$. The game ends when $\mathscr{B}$ halts and outputs a bit b' as its guess of bit b. The corresponding advantage is defined as:

$$\mathbf{Adv}^{\mathrm{srv}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) := 2 \cdot \Pr\left[\, \mathrm{SURV}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \,\right] - 1\,.$$

In addition to the above two notions, BPR specify the following *decryptability* condition.

Definition 3 (Decryptability). *A subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}})$ is said to satisfy (q, δ)-decryptability with respect to the scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ if symmetric encryption scheme $(\widetilde{\mathcal{K}} \times \mathcal{K}, \widetilde{\mathcal{E}}, \mathcal{D}')$ where $\mathcal{D}'((\widetilde{K}, K), C, A, \varrho) := \mathcal{D}(K, C, A, \varrho)$ is (q, δ)-correct (for all choices of inputs i to $\widetilde{\mathcal{E}}$).*

If $\widetilde{\Pi}$ is $(q, 0)$-decryptable with respect to Π for all $q \in \mathbb{N}$, it is said to be perfectly decryptable. We highlight that BPR requires that any subversion satisfies perfect decryptability. For reasons that will become apparent later we chose to distinguish between (q, δ)-decryptability and perfect decryptability. However BPR do not make this distinction and use the term decryptability to mean perfect decryptability.

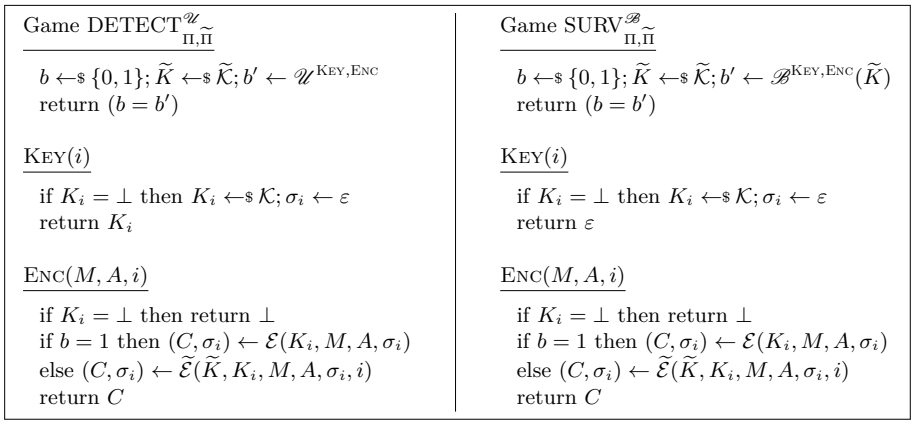

Fig. 2: The DETECT and SURV games from the BPR security model of [4].

OBSERVATIONS. The first thing to note is that the DETECT game is formulated from big brother's point of view who wants his subversion to remain undetected. The notion it yields is that of *undetectability*, and in [4] it is used only for proving *negative* results. For instance BPR use this to show that any randomized encryption scheme can be subverted in an undetectable manner. Concretely, for any randomized scheme Π that uses sufficient amount of randomness there exists a subversion $\widetilde{\Pi}$ such that for all efficient detection tests $\mathscr{U}$ the advantage $\mathbf{Adv}^{\text{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U})$ is small. Moreover, the subversion $\widetilde{\Pi}$ allows big brother to completely recover the user's key K with overwhelming probability.

Security against surveillance is defined through the SURV game. The requirement here is that big brother, who knows the subversion key $\widetilde{K}$, is unable to tell whether ciphertexts are being produced by the real encryption algorithm $\mathcal{E}$ or the subverted encryption algorithm $\widetilde{\mathcal{E}}_{\widetilde{K}}$. This implicitly ensures that if the real scheme is IND-CPA secure then the subverted scheme still does not reveal to big brother anything about the plaintext. Clearly, without any further restriction on $\widetilde{\Pi}$ surveillance resilience is not attainable, since for any scheme Π there always exists a trivial subversion $\widetilde{\Pi}$ and an adversary $\mathscr{B}$ which can distinguish the two. (Consider for example the subversion which appends a redundant zero bit to the ciphertexts.) Hence some resistance to detection should hold simultaneously. This is imposed by means of the decryptability condition. More formally, (in [4]) an encryption scheme Π is said to be surveillance secure if for all subversions $\widetilde{\Pi}$ that are perfectly decryptable with respect to Π and all adversaries $\mathscr{B}$ with reasonable resources its advantage $\mathbf{Adv}^{\text{srv}}_{\Pi,\widetilde{\Pi}}(\mathscr{B})$ is small.

3.2 Critique

In [4], although decryptability is formulated as a correctness requirement, it is really used as a notion of *undetectability*. More precisely, it is understood to be the weakest notion of undetectability that big brother can aim for, and failure

to meet this notion would certainly lead to his subversion being discovered. In fact, BPR write [4, page 6].

> This represents the most basic form of resistance to detection, and we will assume any subversion must meet it.

On the other hand the undetectability notion associated to the DETECT game is meant to be a much stronger one. Another excerpt reads [4, page 7]

> A subversion $\widetilde{\Pi}$ in which this advantage [that is, $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U})$] is negligible for all practical tests $\mathscr{U}$ is said to be *undetectable* and would be one that evades detection in a powerful way. If such a subversion permitted plaintext recovery, big brother would consider it a very successful one.

This all seems to imply that for any subversion, decryptability is a necessary requirement to avoid detection, and that undetectability is sufficient to yield a strong guarantee of avoiding detection. It is hence natural to expect that undetectability implies decryptability, but as the authors of [4] admit this is not the case. The two notions are in fact incomparable. This is a source of inconsistency, especially when considering that the negative and positive results in [4] are established using measures of undetectability that are incomparable.

The main reason for this discord between decryptability and undetectability is that undetectability allows detection test $\mathscr{U}$ to succeed with negligible probability, whereas (perfect) decryptability requires the test's success probability to be exactly zero. This is unnecessarily strict, as detection tests which succeed only with negligible probability are insignificant and pose no effective threat to big brother. Accordingly it is unrealistic to assume that big brother will only produce subversions that satisfy *perfect* decryptability. Requiring the latter imposes an unnatural restriction on big brother's potential subversion strategies, thereby unjustifiably weakening the security notion.

It would seem that both of the above issues could be easily addressed (at least in part) by letting decryptability admit a small negligible error, that is requiring (q, δ)-decryptability, for some small δ, instead of perfect decryptability. In particular, one could hope that decryptability would then be implied by undetectability. Unfortunately the situation is not that simple, and a new problem arises. As we demonstrate next, this minor alteration would render the BPR security notion *unsatisfiable* by any symmetric encryption scheme with a reasonably large message space (e.g., if $|\mathcal{M}| \geq |\mathcal{K}|$). More specifically, for any symmetric encryption scheme we can construct a subversion that not only is (q, δ)-decryptable (with negligible δ for any reasonable value q) but is in fact undetectable, and yet there always exists an adversary $\mathscr{B}$ capable of subverting the scheme. This serves to show that the BPR security definition crucially relies on the presupposition that all subversions must satisfy perfect decryptability, and is consequently a rather fragile security definition.

Theorem 1. *Consider a $(1, \delta)$-correct and ϵ-private symmetric encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with message space $\mathcal{M}$ such that $\{0,1\}^\lambda \subseteq \mathcal{M}$ for some λ*

Algorithm $\widetilde{\mathcal{E}}_{\widetilde{K}}(K, M, A, \sigma, i)$

$(C, \sigma) \leftarrow \mathcal{E}(K, M, A, \sigma)$
if $\mathbf{R}(\widetilde{K}, K, M, A, \sigma, i) = \mathsf{true}$
  then return $(C \parallel K, \sigma)$
else return (C, σ)

Fig. 3: The encryption algorithm of the subversion $\widetilde{\Pi}$ used in Theorem 1.

(for instance, $\lambda = 128$). For any such scheme there exists a subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}})$ that satisfies $(q, q \cdot 2^{-\lambda} + \delta)$-decryptability with respect to Π and $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) \leq q \cdot 2^{-\lambda}$ for all practical detection tests $\mathscr{U}$ making at most q encryption queries. Moreover there exists a corresponding adversary $\mathscr{B}$ such that $\mathbf{Adv}^{\mathrm{srv}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) \geq 1 - (\epsilon + \delta + 2^{-\lambda})$.

Proof. The subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}})$ is defined by letting $\widetilde{\mathcal{K}} := \{0,1\}^\lambda$ and $\widetilde{\mathcal{E}}$ be the algorithm depicted in Fig. 3. The predicate $\mathbf{R}(\widetilde{K}, K, M, A, \sigma, i)$ that is used in $\widetilde{\mathcal{E}}$ takes the boolean value true for all tuples where $\widetilde{K} = M$ and the value false otherwise. Hence note that for all inputs where $\widetilde{K} \neq M$ the subverted encryption algorithm $\widetilde{\mathcal{E}}_{\widetilde{K}}$ behaves exactly like the real encryption algorithm $\mathcal{E}$. Let E denote the event that for some $1 \leq j \leq \ell$ it holds that $\widetilde{K} = \mathbf{M}[j]$. Then for all $1 \leq \ell \leq q$ and all message vectors $\mathbf{M} \in \mathcal{M}^\ell$ we have

$$
\begin{aligned}
&\Pr\left[(\widetilde{K}, K) \leftarrow\!\!\$\ \widetilde{\mathcal{K}} \times \mathcal{K}; (\mathbf{C}, \sigma_\ell) \leftarrow \mathcal{E}_K(\mathbf{M}, \mathbf{A}, \varepsilon); (\mathbf{M}', \varrho_\ell) \leftarrow \mathcal{D}_K(\mathbf{C}, \mathbf{A}, \varepsilon) : \mathbf{M} \neq \mathbf{M}'\right] \\
&\leq \Pr\left[(\widetilde{K}, K) \leftarrow\!\!\$\ \widetilde{\mathcal{K}} \times \mathcal{K} \mid E\right] + \Pr\Big[(\widetilde{K}, K) \leftarrow\!\!\$\ \widetilde{\mathcal{K}} \times \mathcal{K}; \\
&\qquad\qquad (\mathbf{C}, \sigma_\ell) \leftarrow \mathcal{E}_K(\mathbf{M}, \mathbf{A}, \varepsilon); (\mathbf{M}', \varrho_\ell) \leftarrow \mathcal{D}_K(\mathbf{C}, \mathbf{A}, \varepsilon) \,:\, \mathbf{M} \neq \mathbf{M}' \mid \overline{E}\Big] \\
&\leq q \cdot 2^{-\lambda} + \delta\,,
\end{aligned}
$$

where the bound on the second term follows from the δ-correctness of Π. Hence $\widetilde{\Pi}$ satisfies $(q, q \cdot 2^{-\lambda} + \delta)$-decryptability with respect to Π. Since $\mathscr{U}$ is not given any information about $\widetilde{K}$, it is easy to see that for any (even computationally unbounded) detection test $\mathscr{U}$ making at most q queries its advantage $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U})$ is bounded by $q \cdot 2^{-\lambda}$.

The adversary $\mathscr{B}$, which knows the subversion key, simply queries the pair $(\widetilde{K}, A)$ to its encryption oracle for some $A \in \mathcal{AD}$, and gets in return a ciphertext C^*. It then attempts to parse C^* as $C \parallel K$ and checks whether $\widetilde{K} = \mathcal{D}_K(C, A, \varepsilon)$. If this test succeeds it outputs 0 and otherwise it outputs 1. Note that when the encryption oracle is instantiated with the subversion ($b = 0$), the adversary is guaranteed to guess correctly, i.e., outputs 0, with probability $1 - \delta$ by the correctness of Π. Alternatively when the oracle is instantiated with the real scheme ($b = 1$), it can be shown that the decryption test that $\mathscr{B}$ runs on

C^* cannot succeed with probability higher than $\epsilon + 2^{-\lambda}$. Hence, the probability of $\mathscr{B}$ outputting 0 when $b = 1$ is also bounded by this amount. Letting b' denote $\mathscr{B}$'s output and combining the above we have that

$$\begin{aligned}\mathbf{Adv}^{\mathrm{srv}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) &= \Pr[\, b' = 0 \mid b = 0\,] - \Pr[\, b' = 0 \mid b = 1\,] \qquad (1)\\ &\geq 1 - \delta - \epsilon - 2^{-\lambda}\,,\end{aligned}$$

as desired. It only remains to prove the bound on the second term of equation (1). We establish the bound by reducing $\mathscr{B}$ to an IND-CPA adversary $\mathscr{A}$ against Π. The adversary $\mathscr{A}$ starts by picking a subversion key $\widetilde{K}$ uniformly at random and then runs $\mathscr{B}$ on input $\widetilde{K}$. When $\mathscr{B}$ makes its first encryption query (M_0, A), where $M_0 = \widetilde{K}$, $\mathscr{A}$ will sample uniformly at random a second message M_1 of equal length. Then $\mathscr{A}$ submits (M_0, M_1, A) to its own oracle and forwards the ciphertext C^* that the oracle returns to $\mathscr{B}$. At this point $\mathscr{B}$ will halt and $\mathscr{A}$ outputs whatever $\mathscr{B}$ outputs, which we denote by b'. Let d denote the bit in the IND-CPA game indicating which message is being encrypted, then

$$\begin{aligned}\mathbf{Adv}^{\text{ind-cpa}}_{\Pi}(\mathscr{A}) &= 2\Pr\left[\,\text{IND-CPA}^{\mathscr{A}}_{\Pi}\,\right] - 1\\ &= \Pr[\, b' = 0 \mid d = 0\,] - \Pr[\, b' = 0 \mid d = 1\,] \leq \epsilon\,. \qquad (2)\end{aligned}$$

Now note that when C^* corresponds to an encryption of (M_0, A), i.e., $d = 0$, $\mathscr{B}$ gets a perfect simulation of the SURV game with b set to 1. Thus

$$\Pr[\, b' = 0 \mid d = 0\,] = \Pr[\, b' = 0 \mid b = 1\,]\,. \qquad (3)$$

On the other hand when $d = 1$ the ciphertext C^* is independent of M_0, and hence the decryption test that $\mathscr{B}$ runs cannot be better than guessing the value M_0. Therefore

$$\Pr[\, b' = 0 \mid d = 1\,] \leq 2^{-\lambda}\,. \qquad (4)$$

Combining Equations (2),(3) and (4) we get the desired bound:

$$\Pr[\, b' = 0 \mid b = 1\,] \leq \epsilon + 2^{-\lambda}\,.$$

Input-triggered subversions. We emphasize that the above subversion applies generically to any practically relevant symmetric encryption scheme, irrespective of whether it is probabilistic or deterministic and whether it maintains a state or not. Additionally, while we present the subversion of Figure 3 merely as a component of Theorem 1, it actually embodies a powerful subversion strategy[4] for mounting ASAs that are hard to detect. The underlying principle is

[4] This is akin to a trapdoor. It is a classic technique in computer security to introduce trapdoors in various objects and we certainly do not claim to be the first to do so.

that a subversion leaks information to big brother only when receiving specific inputs. That is, in order for big brother to exploit his subversion and undermine the privacy of the communication, a trigger needs to be set. On the other hand, without knowledge of this trigger it is practically impossible to distinguish the subversion from the real scheme. In our case the trigger is the set of inputs for which the predicate **R** holds. In practice, **R** can depend on any information that the subverted encryption algorithm may have access to, such as an IP address, a username, or some location information. Such information, in particular network addresses and routing information, can be readily available in the associated data. It is not unreasonable, and is in fact in conformance with the usual approach adopted in cryptography, to assume that big brother may be capable of influencing this information when it needs to intercept a communication. We hence see no basis for excluding such attacks from consideration.

Security guarantees. BPR start from the premise that surveillance security is not possible without requiring some resistance to detection, and they address this by requiring that all subversions satisfy perfect decryptability. Indeed, it seems that the only way of protecting against ASAs is to have a mechanism to detect such attacks. Accordingly, an encryption scheme should be deemed surveillance secure if we have a sufficiently good chance of detecting subversions of that scheme. However, the BPR security notion gives only a very weak guarantee of detecting ASAs. More specifically, we are only guaranteed to detect a subversion with non-zero probability, regardless of how small that may be. In particular, if for a specific scheme there exist subversions which can all be detected with non-zero but only negligible probability, then in the BPR security model this scheme is considered subversion secure. It should be evident however that such a scheme offers no significant resistance to subversion in practice.

Another shortcoming of relying on decryptability as a means of detection is that it does not clearly state what tests one ought to do in order to detect a subversion. Decryption failures may happen for other reasons, and if they occur sporadically they may easily go unnoticed. Secondly, it may not suffice to rely on the decryption algorithm at the receiver's end. For instance, if ciphertexts contain additional information that big brother can exploit but which would result in a decryption failure, big brother could rectify this at the point of interception after having recovered the information he needs. Alternatively big brother may have replaced the decryption algorithm with one that can handle ciphertexts from the subverted encryption algorithm without raising any exceptions. While for an open system like TLS [7] it may be reasonable to assume that big brother is unable to mount an ASA on all of its implementations, on a closed system[5] there is no reason to assume big brother is not able to substitute both the encryption and decryption algorithms.

[5] This could be some proprietary application/protocol, for which there exists only one implementation, but which uses a standard (non-proprietary) encryption scheme.

4 The Proposed Security Model

The analysis of Section 3.2 leaves us with an unsatisfactory state of affairs. On the one hand we wish for a more realistic security model, devoid of the perfect decryptability condition. On the other hand we saw that this would allow input-triggered subversions which are generically applicable to any symmetric encryption scheme. This in turn raises the question of whether we have any hope at all of protecting against ASAs. We address these questions by proposing an alternative security model which builds on the ideas of Bellare, Paterson and Rogaway [4].

Our premise is that input-triggered subversions cannot be detected with significant probability through a one-time test, as in the DETECT game. Instead, it seems that the best we can hope for is to detect whether the encryption algorithm is leaking information during a communication session. That is we are unable to determine whether the encryption algorithm has been substituted or not, since without knowledge of the trigger we have very little chance of detecting this. However we may be able to detect whether big brother is exploiting the subversion and is able to gather information from it, which is what we really care about.

In formulating security, we consider all possible subversions that big brother may come up with, without imposing any additional restrictions that a subversion must satisfy. Instead we identify a scheme to be subversion resistant, if for all of its possible subversions it hold that either the subversion leaks no information to big brother, or if it does leak information then we can detect it with high probability. We formalize this by means of a second pair of games $\overline{\text{DETECT}}$ and $\overline{\text{SURV}}$. The game $\overline{\text{SURV}}$ is a single-user version of the SURV game from [4], and can be shown to be equivalent, through a standard hybrid argument, up to a factor equal to the number of users. This serves to specify formally what we intuitively referred to as 'leaking information to big brother'. The $\overline{\text{DETECT}}$ game, on the other hand, differs substantially form the DETECT game of the BPR security model. Most importantly, it is intended for specifying a notion of *detectability* rather than *un*detectability. In $\overline{\text{DETECT}}$, the detection test $\mathscr{U}$ does not get access to an encryption oracle, instead it only gets a transcript of $\mathscr{B}$'s queries to its own oracle. We will then quantify the effectiveness of the detection test $\mathscr{U}$ by comparing its success in the $\overline{\text{DETECT}}$ game in guessing the challenge bit to that of $\mathscr{B}$ in the $\overline{\text{SURV}}$ game. This is specified more formally below.

The surveillance game starts by picking a bit b uniformly at random, and then generates the keys K and $\widetilde{K}$. Big brother is then given access to the subversion key and an encryption oracle but not the key K. Depending on the value of b, the encryption oracle will either return encryptions under scheme Π and the user's key K or encryptions under the subverted scheme (which has access to both keys). The adversary outputs a bit b' as its guess of the challenge bit b. See Figure 4 (right) for the details. The detection game is an extension of the surveillance game. First $\mathscr{B}$ is run in the same manner as in the surveillance game and a transcript T of its encryption queries is kept. The detection algorithm $\mathscr{U}$ is then given access to this transcript and the user's key. Its goal is to output a

bit b'' as its guess of the challenge bit b. See Figure 4 for a formal description of both security games. Note that in the $\overline{\text{DETECT}}$ game $\mathscr{B}$'s output is discarded, since it's role in this game is only to generate the transcript of queries.

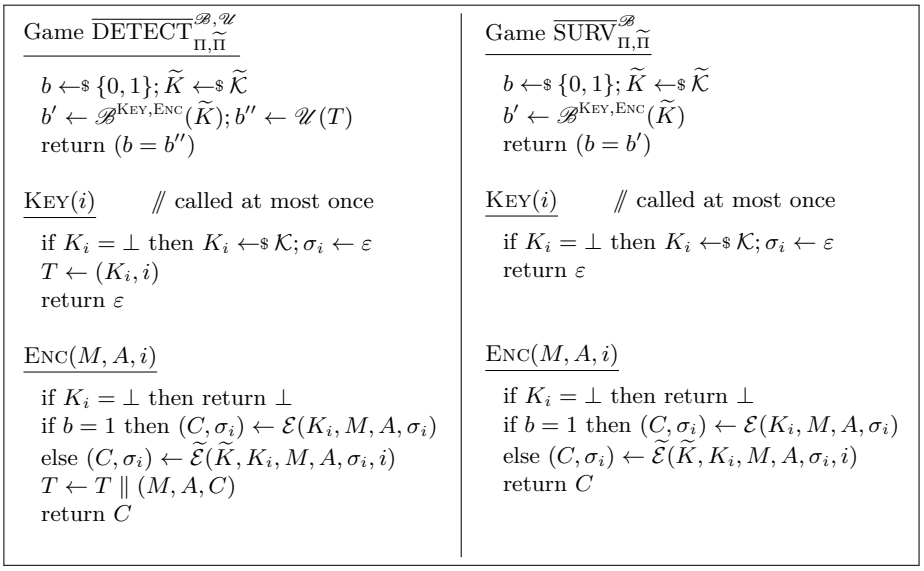

Fig. 4: Games defining the refined single-user security models. Big brother $\mathscr{B}$ can only call the KEY oracle once.

We now move on to define security in terms of the above games. For each of these games we define the corresponding advantages in the usual manner. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}})$ be a subversion of it. For an adversary $\mathscr{B}$ its surveillance advantage is given by:

$$\mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) := 2 \cdot \Pr\left[\, \overline{\text{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \,\right] - 1\,.$$

Similarly, the detection advantage of algorithm $\mathscr{U}$ with respect to $\mathscr{B}$ is given by:

$$\mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B},\mathscr{U}) := 2 \cdot \Pr\left[\, \overline{\text{DETECT}}^{\mathscr{B},\mathscr{U}}_{\Pi,\widetilde{\Pi}} \,\right] - 1\,.$$

We will require that a secure scheme Π come with a detection test $\mathscr{U}$ which will allow the detection of information leakage from a subverted encryption algorithm. Intuitively our security definition should guarantee that for any adversary $\mathscr{B}$ that wins the $\overline{\text{SURV}}$ game with a significant advantage ϵ, the detection test $\mathscr{U}$ will win the $\overline{\text{DETECT}}$ game with a correspondingly significant advantage δ. Thus, as our first attempt at defining subversion resilience we may require that for all adversaries $\mathscr{B}$ and all subversions $\widetilde{\Pi}$ it hold that:

$$\mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B},\mathscr{U}) \leq \delta \implies \mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) \leq \epsilon\,.$$

In the above, we used the contrapositive so that ϵ and δ act as upper bounds rather than lower bounds; of course this is only a matter of taste. For some $\delta, \epsilon \in [0,1]$, let us say that the pair $(\Pi, \mathscr{U})$ is (δ, ϵ)-subversion-resistant if it satisfies the above condition. This definition has the following properties:

(i) A (δ, ϵ)-subversion-resistant pair is also (δ', ϵ')-subversion resistant if $\delta' \leq \delta$ and $\epsilon' \geq \epsilon$.
(ii) No pair can be (δ, ϵ)-subversion-resistant for any (δ, ϵ) where $\delta > \epsilon$.

The first property follows trivially from the definition, whereas the latter property can be shown through the subsequent argument. Assume that $(\Pi, \mathscr{U})$ is a (δ, ϵ)-subversion-resistant where $\delta > \epsilon$. Then there exists a subversion $\widetilde{\Pi}$ and an adversary $\mathscr{B}$ such that

$$\mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}, \mathscr{U}) \leq \delta \text{ and } \mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) > \epsilon\,,$$

thereby reaching a contradiction. More specifically, let $|\widetilde{\mathcal{K}}| \geq (\delta - \epsilon)^{-1}$ and define the subverted encryption algorithm $\widetilde{\mathcal{E}}$ as follows. Split the combined keyspace $\mathcal{K} \times \widetilde{\mathcal{K}}$ into two parts, where $\lfloor \delta \cdot |\mathcal{K}| \cdot |\widetilde{\mathcal{K}}| \rfloor$ of the key pairs belong to the first set and the rest of the key pairs belong to the second set. For key pairs in the first set $\widetilde{\mathcal{E}}$ always returns the special symbol $\perp$, whereas for key pairs in the second set it returns encryptions under $\mathcal{E}$. The Adversary $\mathscr{B}$ asks for an encryption of a fixed message to get C and outputs $(C = \perp)$. Clearly the detection advantage cannot exceed δ, and the surveillance advantage is exactly

$$\frac{\lfloor \delta \cdot |\mathcal{K}| \cdot |\widetilde{\mathcal{K}}| \rfloor}{|\mathcal{K}| \cdot |\widetilde{\mathcal{K}}|} > \delta - \frac{1}{|\widetilde{\mathcal{K}}|} \geq \epsilon\,.$$

While the notion of (δ, ϵ)-subversion-resistance seems reasonable and conformant with the style of concrete security, by itself, it does not yield a satisfactory security definition. To see why, consider a theorem asserting the security of some scheme and a detection test with specific values of δ and ϵ, where good security translates to δ being quantitatively close to ϵ. Then we are guaranteed that when the adversary has an advantage greater than ϵ, the detection algorithm will detect with advantage at least δ. The problem is that such a statement is only useful when the adversary's advantage is greater than but close in value to ϵ. If on the other hand, the surveillance advantage is less than ϵ, then we cannot conclude anything about the success of the detection test. Similarly if the adversary's advantage is much higher than ϵ we are only guaranteed a detection advantage of δ, which is much smaller than the adversary's advantage.

In essence the above definition conveys information about a single point over the range of values which the surveillance advantage can assume, i.e. $[0,1]$. Accordingly a better way to define security is to relate the detection advantage to the surveillance advantage. That is, for some function ρ let us say that the pair $(\Pi, \mathscr{U})$ is ρ-subversion-resistant if for all adversaries $\mathscr{B}$ and all subversions $\widetilde{\Pi}$ it hold that:

$$\mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}, \mathscr{U}) \geq \rho\left(\mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B})\right).$$

Note that from the above statement it is possible to establish a lower bound on the detection advantage (as in the case of (δ, ϵ)-subversion-resistance), for *any* value of the surveillance advantage. It follows from property (i) above, that if such a function ρ exists, it will be monotonic. Moreover, property (ii) says that any function ρ is bounded above by the identity function. Hence, the best security we can hope for in the case of ρ-subversion-resistance, is when ρ is the identity function. As we shall see in the next section, this strong form of ρ-subversion-resistance is achievable and we will adopt it as our proposed security definition.

Definition 4 (Subversion resistance). *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme and let $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}})$ be a subversion of it. For an adversary $\mathscr{B}$ and a detection algorithm $\mathscr{U}$, define the games $\overline{\mathrm{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}}$ and $\overline{\mathrm{DETECT}}^{\mathscr{B},\mathscr{U}}_{\Pi,\widetilde{\Pi}}$ as depicted in Figure 4. The pair $(\Pi, \mathscr{U})$ is said to be subversion resistant if for all adversaries $\mathscr{B}$ and all subversions $\widetilde{\Pi}$ it hold that*

$$\mathbf{Adv}^{\overline{\mathrm{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) \leq \mathbf{Adv}^{\overline{\mathrm{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}, \mathscr{U})\,.$$

Definitional choices. A number of choices have been made in devising the new security definition. Observe that our surveillance game is identical to the single-user version of BPR's original surveillance game in Figure 2.[6] In particular, it allows big brother to launch $\widetilde{K}$-dependent chosen-plaintext attacks. Our detection game is also single-user and this reflects the fact that users do not need to run a coordinated detection procedure. Detection requires the existence of a strong *universal* detector that depends neither on the subverted algorithm nor on big brother. This is in contrast to BPR's formulation, where detection was used for negative results, and non-universal detectors were also allowed. For detection, as in BPR, we assume explicit knowledge of user keys but do not allow access to the (possibly subverted) encryption procedure or the internal state/randomness of the scheme. Weakening the requirements on the detector only strengthens our positive results. On the other hand, the communicated ciphertexts/messages should be made available to the detector. As we have seen, without this strengthening, resistance against input-triggered subversions is impossible even for multi-user oracle-assisted detectors. We note, however, that our actual detection procedure in Section 5 processes ciphertexts one at a time and hence storing only the last computed ciphertext would be sufficient.

5 Subversion Resistance from Unique Ciphertexts

We have not yet determined whether there exist symmetric encryption schemes which satisfy our security definition. In [4] the authors describe a powerful generic

[6] The single-user and multi-user games can be shown equivalent via a standard hybrid argument [4]. Since our detection procedure is also in the single-user setting, we have adopted a single-user surveillance game as well. This choice also translates to a more faithful comparison of concrete advantage terms.

Algorithm $\mathscr{U}(T)$

Parse T as $(K, i) \parallel T'$
$j \leftarrow 1; \mathbf{M} \leftarrow [\,]; \mathbf{A} \leftarrow [\,]; \mathbf{C} \leftarrow [\,]$
for each (M, A, C) in T' do
  $\mathbf{M}[j] \leftarrow M, \mathbf{A}[j] \leftarrow A; \mathbf{C}[j] \leftarrow C$
  $j \leftarrow j + 1$
$(\mathbf{M}', \varrho_\ell) \leftarrow \mathcal{D}_K(\mathbf{C}, \mathbf{A}, \varepsilon)$
return $(\mathbf{M}' = \mathbf{M})$

Fig. 5: The detection test $\mathscr{U}$ used in Theorem 2.

attack, termed the *biased-ciphertext attack*, that can be applied to any probabilistic symmetric encryption scheme. Hence any scheme that resists subversion must be deterministic. Bellare, Paterson, and Rogaway identified the *unique ciphertexts* property for symmetric encryption schemes as sufficient to satisfy their notion of surveillance security. We now show that this property is strong enough to also guarantee subversion security in sense of Definition 4. Let us first recall the definition of unique ciphertexts from [4].

Definition 5 (Unique ciphertexts). *A symmetric encryption scheme* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *is said to have unique ciphertexts if:*

1. Π *satisfies perfect correctness and,*
2. *for all* $\ell \in \mathbb{N}$*, all* $K \in \mathcal{K}$*, all* $\mathbf{M} \in \mathcal{M}^\ell$ *and all* $\mathbf{A} \in \mathcal{AD}^\ell$*, there exists exactly one ciphertext vector* $\mathbf{C}$ *such that:*

$$(\mathbf{M}, \varrho_\ell) \leftarrow \mathcal{D}_K(\mathbf{C}, \mathbf{A}, \varepsilon) \text{ for some } \varrho_\ell\,.$$

It follows from Definition 5 that any symmetric encryption scheme that has unique ciphertexts must be deterministic. Note on the other hand that a deterministic encryption scheme does not necessarily have unique ciphertexts. In [4] it is shown how stateful encryption schemes having unique ciphertexts are easily obtained from most nonce-based encryption schemes [14] which are known to satisfy the tidiness property of [11]. The following theorem says that for schemes with unique ciphertexts we are guaranteed to always detect a subversion with the highest possible success rate.

Theorem 2. *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be a symmetric encryption scheme with unique ciphertexts. Then the detection test* $\mathscr{U}$ *of Figure 5 is such that for all subversions* $\widetilde{\Pi}$ *and all adversaries* $\mathscr{B}$ *we have that*

$$\mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) \leq \mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}, \mathscr{U})\,.$$

Proof. Fix a subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ and an adversary $\mathscr{B}$. Define

Event E: algorithm $\mathscr{B}$ makes a sequence of queries $(\mathbf{M}, \mathbf{A})$ such that the real and subverted encryption algorithms output a different ciphertext sequence, i.e., $\mathcal{E}(K, \mathbf{M}, \mathbf{A}, \varepsilon) \neq \widetilde{\mathcal{E}}(\widetilde{K}, K, \mathbf{M}, \mathbf{A}, \varepsilon, i)$.

Then for any key K, any subversion key $\widetilde{K}$, any subversion $\widetilde{\Pi}$ and any adversary $\mathscr{B}$ the corresponding surveillance advantage can be expressed as:

$$\begin{aligned}\mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) &= 2\Pr\left[\,\overline{\text{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}}\,\right] - 1 \\ &= 2\Pr\left[\,\overline{\text{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \mid E\,\right]\Pr\left[\,E\,\right] + 2\Pr\left[\,\overline{\text{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \mid \overline{E}\,\right]\Pr\left[\,\overline{E}\,\right] - 1\end{aligned}$$

where the probabilities are calculated over the coins of $\mathscr{B}$, the coins of $\widetilde{\mathcal{E}}$, the sampling of the two keys, and bit b. Now if E does *not* occur $\mathscr{B}$ has no information about the bit b in the $\overline{\text{SURV}}$ game, and $\Pr\left[\,\overline{\text{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \mid \overline{E}\,\right] = 1/2$. Hence we may continue

$$\begin{aligned}&= 2\Pr\left[\,\overline{\text{SURV}}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \mid E\,\right]\Pr\left[\,E\,\right] + \Pr\left[\,\overline{E}\,\right] - 1 \\ &\leq \Pr\left[\,E\,\right].\end{aligned}$$

We can expand the detection advantage of $\mathscr{U}$ with respect to $\mathscr{B}$ in a similar manner to obtain:

$$\begin{aligned}\mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B},\mathscr{U}) = 2\cdot\Pr\left[\,\overline{\text{DETECT}}^{\mathscr{B},\mathscr{U}}_{\Pi,\widetilde{\Pi}} \mid E\,\right]\cdot\Pr\left[\,E\,\right] \\ + 2\cdot\Pr\left[\,\overline{\text{DETECT}}^{\mathscr{B},\mathscr{U}}_{\Pi,\widetilde{\Pi}} \mid \overline{E}\,\right]\cdot\Pr\left[\,\overline{E}\,\right] - 1.\end{aligned}$$

As before, if E does not occur $\mathscr{U}$ has no information about the bit b in the $\overline{\text{DETECT}}$ game and cannot do better than guessing. Moreover, when E occurs, it follows from the construction of $\mathscr{U}$ (see Figure 5) and the fact that Π has unique ciphertexts that $\mathscr{U}$ can always distinguish the real scheme from a subversion. Thus $\Pr\left[\,\overline{\text{DETECT}}^{\mathscr{B},\mathscr{U}}_{\Pi,\widetilde{\Pi}} \mid \overline{E}\,\right] = 1/2$ and $\Pr\left[\,\overline{\text{DETECT}}^{\mathscr{B},\mathscr{U}}_{\Pi,\widetilde{\Pi}} \mid E\,\right] = 1$ which yields the desired result:

$$\mathbf{Adv}^{\overline{\text{det}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B},\mathscr{U}) = \Pr\left[\,E\,\right] \geq \mathbf{Adv}^{\overline{\text{srv}}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}).$$

6 Concluding Remarks

Through this work we unravelled definitional challenges in modeling resistance against algorithm substitution attacks (ASA), and in the process we proposed a refinement to address some of the shortcomings of the recent model by Bellare, Paterson, and Rogaway (BPR). Within the new model we are able to re-establish that deploying ciphertext-unique encryption schemes can provide a

provable (but limited) degree of resistance against adversarial entries who carry out ASAs. There are many more aveneues for big brother to undermine the security of real-world cryptosystems than the one considered in [4] and in this work. Characterizing when it is possible to resist against mass surveillance using cryptographic techniques (even in principle) and when this lies beyond the reach of cryptography is in our opinion one of the primary concerns.

Acknowledgments. The authors would like to thank Daniel J. Bernstein for many comments on the earlier versions of the paper. J. P. Degabriele and B. Poettering were supported by EPSRC Leadership Fellowship EP/H005455/1. B. Poettering was also supported by a Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation, and the German Federal Ministry for Education and Research.

References

1. James Ball, Julian Borger, and Glenn Greenwald. Revealed: how US and UK spy agencies defeat internet privacy and security. *The Guardian*, Sep 2013. `http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security`.
2. Elaine Barker and John Kelsey. Recommendation for random number generation using deterministic random bit generators, Jan 2012. `http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf`.
3. Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.
4. Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 1–19, Santa Barbara, USA, August 17–21, 2014. Springer, Germany.
5. Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. On the practical exploitability of dual EC in TLS implementations. In Kevin Fu and Jaeyeon Jung, editors, *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, pages 319–335. USENIX Association, 2014.
6. Claude Crépeau and Alain Slakmon. Simple backdoors for RSA key generation. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 403–416, San Francisco, CA, USA, April 13–17, 2003. Springer, Germany.
7. Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol version 1.2. RFC 5246, August 2008. `https://www.ietf.org/rfc/rfc5246.txt`.
8. Eu-Jin Goh, Dan Boneh, Benny Pinkas, and Philippe Golle. The design and implementation of protocol-based hidden key recovery. In Colin Boyd and Wenbo Mao, editors, *ISC 2003*, volume 2851 of *LNCS*, pages 165–179, Bristol, UK, October 1–3, 2003. Springer, Germany.
9. Glenn Greenwald. *No Place to Hide: Edward Snowden, the NSA and the Surveillance State*. Penguin Books Limited, 2014.

10. Joseph Menn. Exclusive: Secret contract tied NSA and security industry pioneer. *Reuters*, Dec 2013. http://www.reuters.com/article/2013/12/20/us-usa-security-rsa-idUSBRE9BJ1C220131220.
11. Chanathip Namprempre, Phillip Rogaway, and Thomas Shrimpton. Reconsidering generic composition. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 257–274, Copenhagen, Denmark, May 11–15, 2014. Springer, Germany.
12. Nicole Perlroth. Government announces steps to restore confidence on encryption standards. *The New York Times*, Sep 2013. http://bits.blogs.nytimes.com/2013/09/10/government-announces-steps-to-restore-confidence-on-encryption-standards/.
13. Eric Rescorla and Margaret Salter. Extended random values for TLS. Internet Draft, March 2009. https://tools.ietf.org/html/draft-rescorla-tls-extended-random-02.
14. Phillip Rogaway. Nonce-based symmetric encryption. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359, New Delhi, India, February 5–7, 2004. Springer, Germany.
15. Daniel Shurmow and Niels Ferguson. On the possibility of a back door in the NIST SP800-90 dual EC PRNG. CRYPTO Rump Session, 2007. http://rump2007.cr.yp.to/15-shumow.pdf.
16. Gustavus J. Simmons. The prisoners' problem and the subliminal channel. In David Chaum, editor, *CRYPTO'83*, pages 51–67, Santa Barbara, USA, 1983. Plenum Press, New York, USA.
17. Adam Young and Moti Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 89–103, Santa Barbara, USA, August 18–22, 1996. Springer, Germany.
18. Adam Young and Moti Yung. Kleptography: Using cryptography against cryptography. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 62–74, Konstanz, Germany, May 11–15, 1997. Springer, Germany.
19. Adam Young and Moti Yung. The prevalence of kleptographic attacks on discrete-log based cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 264–276, Santa Barbara, USA, August 17–21, 1997. Springer, Germany.
20. Adam Young and Moti Yung. Bandwidth-optimal kleptographic attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 235–250, Paris, France, May 14–16, 2001. Springer, Germany.
21. Adam Young and Moti Yung. Malicious cryptography: Kleptographic aspects (invited talk). In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 7–18, San Francisco, CA, USA, February 14–18, 2005. Springer, Germany.
22. Adam Young and Moti Yung. A space efficient backdoor in RSA and its applications. In Bart Preneel and Stafford Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 128–143, Kingston, Ontario, Canada, August 11–12, 2006. Springer, Germany.
23. Adam L. Young and Moti Yung. Space-efficient kleptography without random oracles. In Teddy Furon, François Cayre, Gwenaël J. Doërr, and Patrick Bas, editors, *Information Hiding, 9th International Workshop, IH 2007, Saint Malo, France, June 11-13, 2007*, volume 4567 of *LNCS*, pages 112–129. Springer, 2007.