A preliminary version of this paper appears in the ACM Computer and Communications Security Conference (CCS), 2015. This is the full version.

Mass-surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks

Mihir Bellare[1] Joseph Jaeger[2] Daniel Kane[3]

April 2017

Abstract

We present new algorithm-substitution attacks (ASAs) on symmetric encryption that improve over prior ones in two ways. First, while prior attacks only broke a sub-class of randomized schemes having a property called coin injectivity, our attacks break *all* randomized schemes. Second, while prior attacks are stateful, ours are stateless, achieving a notion of strong undetectability that we formalize. Together this shows that ASAs are an even more dangerous and powerful mass surveillance method than previously thought. Our work serves to increase awareness about what is possible with ASAs and to spur the search for deterrents and countermeasures.

[1]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-1116800 and CNS-1228890 and a gift from Microsoft. This work was done in part while Bellare was visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant CNS-1523467.

[2]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: jsjaeger@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~jsjaeger/. Supported in part by NSF grants CNS-1116800 and CNS-1228890.

[3]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: dakane@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~dakane/.

Contents

1 Introduction **3**

2 Notation and Definitions **6**

3 ASA Framework **7**

4 Attack and Analysis **10**

5 Defenses **16**

6 Acknowledgements **16**

1 Introduction

The Snowden revelations have exposed a pervasive program of mass surveillance, one of whose potential mechanisms is an algorithm substitution attack (ASA) [19, 20, 3]. In the context of symmetric encryption which is our focus, your encryption code (which implements the prescribed encryption algorithm of the scheme) is replaced by big-brother created malware that aims to undetectably exfiltrate your key through ciphertexts [3].

In this paper we present a new ASA on symmetric encryption that improves over the prior one of BPR [3] in two ways. First, while the prior ASA only broke a sub-class of randomized schemes having a property called coin injectivity, ours breaks *all* randomized schemes. Second, while the prior ASA was stateful, ours is stateless, achieving a stronger notion of undetectability that we motivate and formalize. Together this shows that ASAs are even more dangerous and powerful than previously thought.

ASAs. Alice expects that her encryption code is implementing the encryption algorithm $\mathsf{SE.Enc}$ prescribed by her chosen symmetric encryption scheme SE. In an ASA [3], the code of $\mathsf{SE.Enc}$ has been replaced by big-brother designed malware $\mathsf{A.Enc}$. When Alice calls her encryption routine, giving it key K_{SE} and message m to encrypt, it is $\mathsf{A.Enc}$ that runs, producing a ciphertext c based not only on the given inputs but also on another, hardwired key we denote K_{A}. Colluding with $\mathsf{A.Enc}$ is an accomplice $\mathsf{A.Ext}$ who knows K_{A} and will pick up c by eavesdropping on the channel. $\mathsf{A.Enc}$ aims to produce c in such a way that $\mathsf{A.Ext}$ can violate privacy of Alice's communications, the most effective (from the point of view of big brother) attack being one that recovers K_{SE} from c.

One's first reaction may be that a successful ASA is trivial. For example, let the subverted code $\mathsf{A.Enc}$, given $K_{\mathsf{A}}, K_{\mathsf{SE}}, m$, set ciphertext c equal to the encryption key K_{SE}. BPR [3] argue that this is attack is unattractive to big brother because it is *detectable*. Anyone (in particular the decrypter), obtaining c, will see that something is not right. BPR [3] put forth the goal of an ASA as being to violate privacy while remaining undetectable.

Undetectability. To rigorously explore ASAs, BPR [3] provide a formalization of undetectability. Their viewpoint is that the detecter is the decrypter. It thus knows K_{SE} and can check that a ciphertext decrypts to an intended message. But it can do other checks as well. The definition models detecter $\mathscr{D}$ as an adversary given input K_{SE} and access to an encryption oracle Enc that, given a message m, returns ciphertext c computed either via $\mathsf{SE.Enc}$ on inputs K_{SE}, m or via $\mathsf{A.Enc}$ on inputs $K_{\mathsf{A}}, K_{\mathsf{SE}}, m$. Undetectability requires that no efficient $\mathscr{D}$ should be able to tell which.

The BPR attack. The question that emerges is whether it is possible to mount a successful but undetectable ASA. BPR [3] address this in depth when $\mathsf{SE.Enc}$ is randomized. (Here and henceforth, a randomized scheme means a stateless scheme with non-trivial entropy in the ciphertexts. See Section 4 for a formal definition. A deterministic scheme, which has no entropy in the ciphertexts, does not qualify!) Randomized schemes are the most basic and common, making this question important. Their strongest result is represented by their *biased ciphertext attack*. We will first explain the attack and then discuss our work.

Let us view $\mathsf{SE.Enc}$ as a deterministic function of K_{SE}, m and coins r, producing a ciphertext as $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$. In the true encryption process, r is chosen at random. In the biased ciphertext attack of BPR [3], $\mathsf{A.Enc}$ also produces c as $\mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$ but picks r, not uniformly at random, but at random from a subset S of the set of all possible coins, where S is defined as the set of r so that a PRF F under K_{A}, when applied to input $c = \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$, returns the first bit $K_{\mathsf{SE}}[1]$ of K_{SE}. This allows $\mathsf{A.Ext}$, given K_{A}, c, to retrieve $K_{\mathsf{SE}}[1] = \mathsf{F}(K_{\mathsf{A}}, c)$. The process is repeated to exfiltrate K_{SE} bit by bit.

The difficulty is assessing undetectability. The ciphertext c produced by $\mathsf{A.Enc}$ is not distributed the same way as one produced by the true encryption process. Can a detection adversary $\mathscr{D}$ tell? This question turns out to be surprisingly difficult. BPR [3] provide a partial answer. Calling a scheme SE coin injective if the mapping $r \mapsto \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$ is injective for all K_{SE}, m, they use a combinatorial analysis [3, Lemma 1] to prove undetectability in this case. Based on this they move away from randomized, stateless schemes to obtain security via deterministic, stateful ones.

Shortcomings of prior work. The biased ciphertext attack has two shortcomings. Both are noted in BPR [3] and left as open problems which we will resolve.

The first shortcoming is that the attack only applies to coin injective schemes. The basic question left un-answered here is, does there exist an ASA-secure randomized scheme, or are *all* randomized schemes subject to attack?

The second shortcoming is that in the biased ciphertext ASA, the subverted encryption algorithm $\mathsf{A.Enc}$ is *stateful*, maintaining state σ across invocations. (The state in this case is an integer representing either which bit of the key $\mathsf{A.Enc}$ is trying to exfiltrate or taking a special value to indicate that exfiltration is complete and encryption should be be done as usual.) This renders the attack detectable in practice in simple ways. For example, a state reset, as can happen with a reboot or cloning to create a virtual machine, leads, in their attack, to detection. However, this is not captured by their definition of detectability, under which they prove their attack undetectable (for coin injective schemes). BPR [3] recognize this and comment that a stateless attack would be better, but that they do not know how to make their attack stateless.

Contributions in brief. We contribute (1) new definitions (2) new attacks and (3) new analyses. The definitions are for strong undetectability (which captures the above issues and can only be met by stateless schemes) and key recovery security, the latter reflecting the need to formalize not just security goals but attack goals. The attack is a new ASA shown to achieve both. The analyses establishing this resolve technical issues from BPR [3] via a different approach. As a by product, we resolve the above questions, presenting a stateless ASA that breaks all randomized encryption schemes and showing that the move to deterministic schemes for security, made by BPR [3], is a necessary one.

Strong undetectability. Our first contribution is to introduce and formalize a stronger notion of undetectability called strong undetectability. Our formalization, in Section 3, uses the framework of the above discussed definition of BPR [3] but makes the oracle Enc more powerful with regard both to inputs and outputs. It now takes input not just a message, but a key, allowing the detector to pick, rather than merely know, the encryption key K_{SE}. It returns not just the ciphertext, but the current state σ of the encryption code.

The last means that as long as the good encryption algorithm $\mathsf{SE.Enc}$ is stateless (the case we are interested in here), statelessness of $\mathsf{A.Enc}$ is a necessary (but not sufficient) condition for strong undetectability. That is, a strongly undetectable ASA must be stateless. A consequence of this is that the ASA will not be detectable upon system reset or cloning, meaning strong undetectability excludes the easy and natural methods of detection allowed by undetectability under BPR [3].

Ensuring this was the main purpose of the definition. Beyond that, while BPR [3] took the view that detection is performed by the decrypter, strong undetectability suggests that detection may to some extent be performed by the encryptor. It imagines that the detector may have blackbox access to the encryption code and can experiment with it, so that it can feed it keys of its choice and see whether or not it maintains state. However one must be careful to not take this interpretation to an extreme, for there are certainly detection methods that someone with blackbox access to the code could employ that strong undetectability does not capture, for example timing the responses and comparing this to the time the real algorithm would be expected to take. Indeed, it is not

possible to evade *all* forms of detection. We are trying to evade more forms of detection than prior work and figure out how "undetectable" a successful ASA can be.

The biased ciphertext ASA of BPR [3] will not be strongly undetectable due to its use of non-trivial state. The question this raises is whether strongly undetectable (and in particular stateless) ASAs are possible, and on which randomized encryption schemes. We will show that they are possible on all randomized schemes.

Key recovery. Beyond introducing and formalizing strong undetectability, we take a more rigorous approach to attack goals. As is conventional in cryptography, BPR [3] formalize security goals, not attacker goals. Thus, they provide an indistinguishability style surveillance advantage, viewing the goal as being to ensure that this advantage stays small for undetectable ASAs. This is a good goal for the scheme designer to achieve, but its violation by an attacker means little. Big brother is after more, specifically recovery of the target key. We carefully formalize a framework for key recovery attacks in which we can compare different ones to differentiate between the effectiveness of different attacks and make rigorous claims about what our attacks achieve.

Our formalization introduces as a parameter a message distribution $\mathcal{M}$, distinct from the adversary, that represents the messages encrypted by the encrypter. The larger the class of message distributions for which the attack succeeds, the more powerful the attack. Our ASA will succeed for *all* message distributions, which means that big brother recovers the target key regardless of what messages the sender sends. In particular, big brother would not need to pick or control the messages to succeed.

ASA. We now sketch our new ASA which we will show to achieve key recovery (for all message distributions) and strong undetectability for all randomized encryption schemes. Let the good key K_{SE} have length k, and let PRF F take outputs in the set $\{0,1\} \times [k]$ where $[k] = \{1, \ldots, k\}$, meaning an output is a pair (v, i) where v is a bit and i is an index into the key. In our new ASA, the subverted encryption algorithm $\mathsf{A.Enc}$, given $K_{\mathsf{A}}, K_{\mathsf{SE}}, m$, aims to pick r such that the (v, i) returned by $\mathsf{F}(K_{\mathsf{A}}, \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r))$ satisfies $v = K_{\mathsf{SE}}[i]$, returning $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$. $\mathsf{A.Ext}$ can retrieve $(K_{\mathsf{SE}}[i], i) = \mathsf{F}(K_{\mathsf{A}}, c)$. In this way it gets the i-th bit of the key.

However, there are some difficulties with the above description. First, it may not be possible to directly and efficiently pick r in the way described. In our ASA, which is presented in detail in Section 4, $\mathsf{A.Enc}$ uses rejection sampling, repeatedly picking r at random until either the desired condition is met or it times out, when the latter happens being controlled by a parameter s of the attack. Second, we must use a series of q encryptions to allow $\mathsf{A.Ext}$ to recover the entire key, so that we must determine how to pick q, which is another parameter of the attack. The challenge is to prove both key recovery and strong undetectability without making any assumptions on the encryption scheme other than that it is randomized, and with good, concrete bounds enabling concrete and practical choices of q, s that make the attack both efficient and effective. The analyses we discuss next resolve these challenges.

Analyses. Theorem 4.1 proves that big brother, with our ASA, will indeed recover the target key, with high probability even for relatively small values of the attack parameters q, s. Given q, s and the key length k of the targeted randomized encryption scheme SE, the theorem gives a concrete lower bound on the key recovery advantage of our adversary as a function of q, s, k. The proof is a sequence of games. We begin, in a standard way, by exploiting the assumed PRF security of F to move to a game where it is replaced by a random function. We then exploit the assumption that the scheme is randomized (ciphertexts have non-trivial min-entropy) to move to a game where the (v, i) values are all picked independently at random. This allow us to move to a game where the sampling continues infinitely, reaching the conceptually correct distribution of the above attack idea. A coupon collector argument provides the bound for the last game. Each game transition

accumulates an error, the sum of these being our final bound. We show with a numerical example that a 128 bit key is recovered quite quickly.

Our approach to proving (strong) undetectability is entirely different from that used by BPR [3] to prove undetectability of their biased ciphertext attack, and it is this novel analysis that allows us to avoid extra assumptions like coin injectivity on the encryption scheme SE. While they use a combinatorial analysis [3, Lemma 1], we use a game sequence. Theorem 4.2 proves strong undetectability of our ASA assuming only that SE is randomized (ciphertexts have non-trivial min entropy) and F is a PRF.

Related work. Simmons work on subliminal channels [15, 16, 17, 18] was an early indication of the danger of ASAs. ASAs are part of the broader framework of kleptography studied by Young and Yung [19, 20, 21, 23, 22]. Their ideas seem prescient now. Back-doored blockciphers were studied in [12, 10, 11]. Goh, Boneh, Pinkas and Golle [7] show how to add key recovery to the SSL/TLS and SSH protocols. Cryptographic reverse firewalls [9] represent an architecture to counter ASAs via trusted code in network perimeter filters. Dodis, Ganesh, Golovnev, Juels and Ristenpart [6] provide a formal treatment of backdooring of PRGs, another form of subversion. Russell, Tang, Yung and Zhou [13] consider ASAs on one-way and trapdoor one-way functions. The survey by Schneier, Fredrikson, Kohno and Ristenpart [14] takes a broader look at subversion, providing useful categorizations. In independent work that we will discuss in more depth after giving our definitions, Degabriele, Farshim and Poettering [5] critique and refine the definitions of BPR [3]. Ateniese, Magri and Venturi [1] study ASAs on signature schemes in particular giving stateful attacks based on the methods of BPR [3]. We note that our methods can be employed to make their attacks stateless as well.

2 Notation and Definitions

Notation. If n is an integer then we let $[n] = \{1, \ldots, n\}$. If x is a string then $|x|$ denotes its length while if S is a set then $|S|$ denotes it size. By ε we denote the empty string. By $x[i]$ we denote the i-th bit of a string x, for $i \in [|x|]$. By $s \leftarrow\$ S$ we denote picking s at random from set S. If A is a randomized algorithm then $y \leftarrow A(x_1, \ldots; r)$ denotes running A on inputs $x_1, \ldots$ and coins r to deterministically obtain output y, and $y \leftarrow\$ A(x_1, \ldots)$ denotes picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. Definitions and proofs use code-based games [4]. See Fig. 1 for an example of a game. If G is a game then Pr[G] denotes the probability that it returns true. We adopt the convention that the running time of an adversary means the worst case execution time of the adversary in the game that executes it, so that time for game setup steps and time to compute answers to oracle queries is included.

PRFs. We recall the definition as per [2, 8]. Let $\mathsf{F} : \{0,1\}^\ell \times \{0,1\}^* \to R$ be a function taking a key $L \in \{0,1\}^\ell$ and input $c \in \{0,1\}^*$ to return an output $\mathsf{F}(L, c) \in R$. Consider game $\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}}$ associated to F and adversary $\mathscr{F}$. It provides the adversary with an oracle Fn as shown. Let

$$\mathbf{Adv}_{\mathsf{F}}^{\mathrm{prf}}(\mathscr{F}) = 2\Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}}] - 1$$

be the prf advantage of adversary $\mathscr{F}$ against function F. In proofs we will use the standard fact that it can also be expressed as

$$\begin{aligned} &\mathbf{Adv}_{\mathsf{F}}^{\mathrm{prf}}(\mathscr{F}) \\ &= \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \mid b_{\mathrm{prf}} = 1] - (1 - \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \mid b_{\mathrm{prf}} = 0]) \ . \end{aligned} \tag{1}$$

Game $\mathrm{PRF}^{\mathscr{F}}_{\mathsf{F}}$

$L \leftarrow\$ \{0,1\}^{\ell}$; $b_{\mathrm{prf}} \leftarrow\$ \{0,1\}$; $C \leftarrow \emptyset$

$b'_{\mathrm{prf}} \leftarrow\$ \mathscr{F}^{\mathrm{FN}}$

Return $(b_{\mathrm{prf}} = b'_{\mathrm{prf}})$

$\mathrm{FN}(c)$

If $(b_{\mathrm{prf}} = 1)$ then $y_c \leftarrow \mathsf{F}(L, c)$

Else

  If $c \notin C$ then $y_c \leftarrow\$ R$

  $C \leftarrow C \cup \{c\}$

Return y_c

Figure 1: Game used to define prf advantage of adversary $\mathscr{F}$ against function F.

3 ASA Framework

We recall basic syntax of symmetric encryption. We then provide our novel definitions for ASAs, namely strong undetectability as well as adversary advantage in key recovery.

Symmetric encryption. A (symmetric) encryption scheme SE specifies the following. Via $K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$, one selects a key of length $\mathsf{SE.kl}$. Encryption algorithm $\mathsf{SE.Enc}$ takes K_{SE}, message m and coins $r \in \{0,1\}^{\mathsf{SE.rl}}$ to deterministically obtain ciphertext $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$. By $c \leftarrow\$ \mathsf{SE.Enc}(K_{\mathsf{SE}}, m)$ we denote the operation $r \leftarrow\$ \{0,1\}^{\mathsf{SE.rl}}$; $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$. Decryption algorithm $\mathsf{SE.Dec}$ is deterministic and we require that

$$\mathsf{SE.Dec}(K_{\mathsf{SE}}, \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)) = m$$

for all K_{SE}, m, r.

ASAs. An *algorithm substitution attack* (ASA) A specifies the following. Via $K_{\mathsf{A}} \leftarrow\$ \{0,1\}^{\mathsf{A.kl}}$, one selects a key of length $\mathsf{A.kl}$. The subverted encryption algorithm $\mathsf{A.Enc}$ takes $K_{\mathsf{A}}, K_{\mathsf{SE}}, m$ and current state σ to produce ciphertext c and updated state, $(c, \sigma) \leftarrow\$ \mathsf{A.Enc}(K_{\mathsf{A}}, K_{\mathsf{SE}}, m, \sigma)$. (Denoting the state the same in input and output simply indicates an update of this variable.) The idea is that A is specified by big brother. Key K_{A} is shared between the subverted encryption algorithm $\mathsf{A.Enc}$ and its external accomplice $\mathsf{A.Ext}$ who aims, from subverted ciphertexts, to violate security of SE. Note that conceptually, A is an adversary (the ASA) rather than a scheme, a departure in perspective from BPR [3]. We say that $\mathsf{A.Enc}$ is stateless if the updated state it returns is always the empty string ε, meaning its output for any inputs $K_{\mathsf{A}}, K_{\mathsf{SE}}, m, \sigma$ has the form (c, ε). In this case we might drop σ in both input and output, writing $c \leftarrow\$ \mathsf{A.Enc}(K_{\mathsf{A}}, K_{\mathsf{SE}}, m)$.

We said that A's goal is to violate security of SE. With no further conditions, this is too easy, as explained in Section 1. The goal for big brother is an attack that is undetectable yet violates security of SE. We now turn to formalizing each of these components.

Strong undetectability. Consider game SDET of Fig. 2 associated to encryption scheme SE, ASA A and a detection adversary $\mathscr{D}$. It can obtain via oracle Enc encryptions of messages of its choice under keys of its choice, produced either via $\mathsf{SE.Enc}$ (when $b = 1$) or via $\mathsf{A.Enc}$ (when $b = 0$), and aims from examination of these ciphertexts to determine b. Let

$$\mathbf{Adv}^{\mathrm{sdet}}_{\mathsf{SE},\mathsf{A}}(\mathscr{D}) = 2\Pr[\mathrm{SDET}^{\mathscr{D}}_{\mathsf{SE},\mathsf{A}}] - 1 \ .$$

This is $\mathscr{D}$'s advantage in detecting the subversion.

Game $\mathrm{SDET}^{\mathscr{D}}_{\mathsf{SE},\mathsf{A}}$

$K_{\mathsf{A}} \leftarrow\$ \{0,1\}^{\mathsf{A.kl}}$
$b \leftarrow\$ \{0,1\}$; $\sigma \leftarrow \varepsilon$; $b' \leftarrow\$ \mathscr{D}^{\textsc{Enc}}$
Return $(b = b')$

$\textsc{Enc}(K, m)$
If $(b = 1)$ then $c \leftarrow\$ \mathsf{SE.Enc}(K, m)$
Else $(c, \sigma) \leftarrow\$ \mathsf{A.Enc}(K_{\mathsf{A}}, K, m, \sigma)$
Return (c, σ)

Game $\mathrm{KR}^{\mathsf{A}}_{\mathsf{SE},\mathcal{M}}$

$K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$; $K_{\mathsf{A}} \leftarrow\$ \{0,1\}^{\mathsf{A.kl}}$; $\sigma \leftarrow \varepsilon$; $\sigma' \leftarrow \varepsilon$
$K \leftarrow\$ \mathsf{A.Ext}^{\textsc{Enc}}(K_{\mathsf{A}})$; Return $(K_{\mathsf{SE}} = K)$

$\textsc{Enc}()$
$(m, \sigma') \leftarrow\$ \mathcal{M}(\sigma')$
$(c, \sigma) \leftarrow\$ \mathsf{A.Enc}(K_{\mathsf{A}}, K_{\mathsf{SE}}, m, \sigma)$
Return c

Figure 2: Games used to define detection and key recovery security of subversion A of encryption scheme SE.

The state σ returned by $\textsc{Enc}$ will always be the empty string ε in the case $b = 1$ because $\mathsf{SE.Enc}$ is stateless. Thus, if $\mathscr{D}$ ever sees $\sigma \neq \varepsilon$ in a reply to a $\textsc{Enc}$ query, it knows that $b = 0$, meaning the subversion has been detected. Our formalization thus effectively implies that a subversion must be stateless to be undetectable. A consequence of this is that state reset, as can arise due to system reboot or cloning to create a virtual machine, will not allow the ASA to be detected, unlike for undetectability as per BPR [3]. This exclusion of some simple and natural forms of detection not covered by BPR [3] is the main consequence and intent of the new definition.

However there are other directions as well. The view of BPR [3] was that detection is performed by the decryptor. Detection through state reset continues that perspective, for this can in many cases lead to detection from the ciphertexts alone. But strong undetectability also moves towards a perspective where the encryptor, not just the decryptor, may be making some attempts at detection. We view the detector as having some sort of blackbox access to the encryption code, so that it can pick inputs and see all outputs written to memory. In particular, the detector not just knows, but can pick, the encryption key K, and it can see any state σ that the encryption code tries to maintain across invocations. Thus $\mathscr{D}$ can supply, in calls to $\textsc{Enc}$, not just the message, but *any* key $K \in \{0,1\}^{\mathsf{SE.kl}}$ of its choice for $\mathsf{SE.Enc}$, getting in response not just the ciphertext c, but also σ. This obviously significantly increases the power of the detector.

However we must be careful to note that we do not capture *all* possible detection strategies that a "real" detector in such a position could mount. For example the detector could measure the CPU time of an execution of the code and compare this with the expected CPU time of the real code. Or it could look at the number of calls to the underlying pseudo-random number generator that is being used to obtain coins. These and other detection methods are not covered by our definition.

The fact is that it impossible for an ASA to evade *all* forms of detection. Our work aims to understand how far we can push the boundary. Security in this domain is a tradeoff between

detection effort and attack success. Our results indicate that detecting ASAs takes more effort than was previously thought.

We note that strong undetectability implies the BPR notion of undetectability. A proof may be given by reduction, the idea being as follows. Recall that in the detection game of BPR [3], the undetectability adversary $\mathscr{D}'$ has access to an oracle KEY that, given i, returns a key $K_i \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$ for user i, and an oracle ENC that, given message M and i, returns a ciphertext produced by running either $\mathsf{SE.Enc}$ or $\mathsf{A.Enc}$ to encrypt M under K_i, the choice depending on the challenge bit that $\mathscr{D}'$ is trying to guess. Given such a $\mathscr{D}'$, we build a strong undetectability adversary $\mathscr{D}$ with strong undetectability advantage at least the undetectability advantage of $\mathscr{D}'$. Our adversary $\mathscr{D}$ runs $\mathscr{D}'$. When the latter makes a query $\textsc{Key}(i)$, adversary $\mathscr{D}$ itself picks $K_i \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$ and returns it to $\mathscr{D}'$. When the latter makes a query $\textsc{Enc}(M,i)$, adversary $\mathscr{D}$ queries its own ENC oracle with K_i, M to get back (c,σ) and returns c to $\mathscr{D}'$. Finally $\mathscr{D}$ returns the same decision as $\mathscr{D}'$.

We note that the detectability game of BPR was explicitly multi-user, meaning it involved multiple keys. The strong undetectability game instead allows the adversary to query ENC with any key of its choice. One advantage of this formulation, as we have just illustrated, is that muti-user security is a consequence.

KEY RECOVERY. The other side of the coin is what it means for an ASA to succeed, meaning violate the security of the users of SE. One measure, formalized in BPR [3], is distinguishability, meaning A succeeds if it can distinguish encryptions of messages of its choice. Another measure is key recovery, meaning A succeeds if it finds K_{SE}. Distinguishing is a strong measure for security but a weak one for attacks (achieving it provides high security, but violating it entails little loss) while key recovery is a weak measure for security but a strong one for attacks (violating it is very damaging but achieving it means little for security). Since our focus is attacks, we target and formalize key recovery.

Game KR of Fig. 2 is parameterized by a message sampler algorithm $\mathcal{M}$ that, given its current state σ' returns the next message m to be encrypted and updated state. It represents the choice of messages made by the sender. A wins if $\mathsf{A.Ext}$ recovers the key K_{SE} from the ciphertexts produced by the subverted encryption algorithm $\mathsf{A.Enc}$ on messages produced by $\mathcal{M}$. (The state σ maintained by the latter is entirely distinct from the state σ' of $\mathcal{M}$.) The key recovery advantage of A is

$$\mathbf{Adv}^{\mathrm{kr}}_{\mathsf{SE},\mathcal{M}}(\mathsf{A}) = \Pr[\mathrm{KR}^{\mathsf{A}}_{\mathsf{SE},\mathcal{M}}] \ .$$

Parametrization by $\mathcal{M}$ allows a fine-grained taxonomy of key recovery attacks. The less they assume about $\mathcal{M}$, the stronger they are. The strongest attack is one that works for any $\mathcal{M}$. This corresponds to an attack that works regardless of what messages the encrypter chooses to encrypt. A somewhat weaker attack might work for certain message sequences, meaning some restricted class of samplers $\mathcal{M}$. Our ASA is of the stronger type.

DISCUSSION AND EXTENSIONS. Game SDET of Fig. 2 has been written for the case where encryption scheme SE is stateless, since this is the case of interest for our attacks. It can be extended to the case where SE is stateful. In this case, when $b = 1$, algorithm $\mathsf{SE.Enc}$ takes the current state σ as an additional input and returns not just ciphertext c but also an updated state σ. Additionally one should provide the adversary with a reset oracle RESET that resets the state σ to ε. The notion is thus requiring that the detector cannot tell whether it is talking to the real or subverted encryption algorithm even if it sees the state and can reset it. Note that the reset oracle is redundant (and hence has been omitted) in our present context because with stateless schemes the definition already implies that a subversion must be stateless as well, and with both stateless a reset oracle is vacuous.

BPR [3] had required any ASA A to satisfy a decryptability condition which asked that ciphertexts produced under $\mathsf{A.Enc}(\cdot, K, \cdot, \cdot)$ decrypted correctly under $\mathsf{SE.Dec}(K, \cdot)$. We have dropped this condition, so that decryptability holds only to the extent that it is implied by strong undetectability, which we think is more realistic from a detection perspective. However, the ASAs we provide do meet the decryptability condition of BPR [3].

DFP [5] suggest that the BPR decryptability condition is too strong. They suggest a relaxation, and then provide an ASA that is undetectable under the BPR definition, meets their relaxed decryptability condition, and yet succeeds in that it violates the BPR subversion security definition. However the DFP attack does not succeed in violating our key recovery notion, and from a practical perspective, is weak. In the DFP attack, $\mathsf{A.Enc}(K_{\mathsf{A}}, K, m, \sigma)$ returns K when $m = K_{\mathsf{A}}$. That is, the attack requires that the attacker can induce the encrpytor to encrypt the message $m = K_{\mathsf{A}}$. But in practice this is quite hard and it is unlikely that a "real" sender will ever encrypt this message. Our attacks are much more powerful since the attacker succeeds regardless of what messages the encryptor encrypts. This is captured by the introduction of $\mathcal{M}$ in our key recovery definition and the fact that the attack succeeds regardless of its choice. In fact in our key recovery definition, the messages being encrypted in the attack cannot even depend on the subverter's key K_{A}, since the latter is not given to $\mathcal{M}$. The latter is why the DFP attack [5] does not succeed under our key recovery definition.

The issues here reflect, as we discussed above, that achieving a certain notion may provide good practical security, but violating the same notion may not constitute an effective practical attack. (That is, insecurity is not really the opposite of security. There is a lot in between.) Thus, achieving security under the BPR subversion notion is a good target for a scheme. But violating security under this notion need not be (and, in the case of DFP [5], is not) an effective attack. Conversely, our key recovery notion is a good one for attacks but not a good one for security: violating it constitutes a powerful attack, but achieving it provides only weak security. As an analogy, indistinguishability (semantic security) is accepted as a good security target for encryption schemes that one wants constructions to reach, but violating it is a weak attack.

4 Attack and Analysis

We present our ASA, having the following properties: (1) It is stateless and proven *strongly* undetectable (2) It breaks *any* given encryption scheme that has a non-trivial amount of randomization, and (3) break here is in the strong sense of key recovery for arbitrary message distributions. The attack is simple to specify but the analysis is more challenging. We provide a careful analysis to establish both key recovery and strong undetectability with concrete bounds. From these results we can extract concrete values of the parameters for a concrete attack.

<u>Preliminaries.</u> As we discussed, we need SE to have non-trivial randomization. (Otherwise, a deterministic scheme is just a special case of a randomized scheme.) Formally, this means that ciphertexts have some min-entropy. To measure this we define the min-entropy $\mathbf{H}_\infty(\mathsf{SE})$ of the scheme SE via

$$2^{-\mathbf{H}_\infty(\mathsf{SE})} = \max_{K_{\mathsf{SE}}, m, c} \Pr[\mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r) = c]$$

Here, with K_{SE}, m, c fixed, the probability is over a random choice of r from $\{0,1\}^{\mathsf{SE.rl}}$. Our results will assume that $2^{-\mathbf{H}_\infty(\mathsf{SE})}$ is negligible. We note that we could use collision entropy in place of min entropy.

<u>Attack description.</u> Let SE denote the target encryption scheme and let $F\colon \{0,1\}^{\mathsf{F.kl}} \times \{0,1\}^* \to$

$\underline{\mathsf{A.Enc}(K_{\mathsf{A}}, K_{\mathsf{SE}}, m, \sigma)}$

$j \leftarrow 0$
Repeat
  $j \leftarrow j + 1$
  $r \leftarrow\$ \{0,1\}^{\mathsf{SE.rl}}$
  $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$
  $(v, t) \leftarrow \mathsf{F}(K_{\mathsf{A}}, c)$
  $\mathsf{success} \leftarrow (K_{\mathsf{SE}}[t] = v)$
  $\mathsf{out\text{-}of\text{-}time} \leftarrow (j = s)$
Until (success OR out-of-time)
Return (c, ϵ)

$\underline{\mathsf{A.Ext}(K_{\mathsf{A}})}$

$K \leftarrow 0^{\mathsf{SE.kl}}$
For $i = 1, \ldots, q$ do
  $c \leftarrow\$ \textsc{Enc}()$
  $(v, t) \leftarrow \mathsf{F}(K_{\mathsf{A}}, c)$
  $K[t] \leftarrow v$
Return K

Figure 3: ASA A for encryption scheme SE has algorithms $\mathsf{A.Enc}$ and $\mathsf{A.Ext}$ as shown above. Here $\mathsf{F}\colon \{0,1\}^{\mathsf{A.kl}} \times \{0,1\}^* \to \{0,1\} \times [\mathsf{SE.kl}]$ is a PRF used in the attack and $q, s \geq 1$ are parameters of the attack.

$\{0,1\} \times [\mathsf{SE.kl}]$ be a PRF and key length $\mathsf{A.kl} = \mathsf{F.kl}$, meaning K_{A} will be a key for the PRF F defining the function $f(\cdot) = \mathsf{F}(K_{\mathsf{A}}, \cdot)$. The algorithms $\mathsf{A.Enc}$ and $\mathsf{A.Ext}$ are shown in Fig. 3. Here $s, q \geq 1$ are parameters of the attack. (We will show that quite small values of these suffice.)

Subverted encryption algorithm $\mathsf{A.Enc}$ is given $K_{\mathsf{A}}, K_{\mathsf{SE}}$, message m and a state σ that it ignores. (It will be stateless and thus it will always be that $\sigma = \varepsilon$.) Its goal is to pick r so that if $c = \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$ and $(v, t) = \mathsf{F}(K_{\mathsf{A}}, c)$ then $K_{\mathsf{SE}}[t] = v$. It would then return c. When $\mathsf{A.Ext}$ picks up c, it can compute $(v, t) = F(K_{\mathsf{SE}}, c)$ and set $K[t] = v$. An appropriate choice of q will ensure that $\mathsf{A.Ext}$ eventually gets all bits of the key, meaning $K = K_{\mathsf{SE}}$ except with tiny probability.

Conceptually, then, we imagine $\mathsf{A.Enc}$ as trying to pick r at random subject to the constraint that if $c = \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$ and $(v, t) = \mathsf{F}(K_{\mathsf{A}}, c)$ then $K_{\mathsf{SE}}[t] = v$. However it is not clear how to directly pick r in this way. Indeed, such an r may not even exist. To resolve this, $\mathsf{A.Enc}$ in Fig. 3 samples by picking r at random from the full space $\{0,1\}^{\mathsf{SE.rl}}$ until either r satisfies the desired condition (at which point the flag $\mathsf{success}$ becomes true) or the process exceeds the number of allowed sampling attempts s (at which point the flag $\mathsf{out\text{-}of\text{-}time}$ becomes true). The tradeoff is that for efficiency we want s to be quite small but the smaller it is the farther is the distribution of r from the conceptually desired one. The theorems and analyses that follow will deal with these issues and show how to appropriately pick both s and q for an effective yet efficient attack.

<u>Key recovery.</u> The following theorem lower bounds the key recovery advantage of our ASA A of Fig. 3, showing that it is close to one for reasonable values of the parameters, meaning the attack can efficiently and successfully recover the target key.

We recall that our convention is that adversary running time refers to the time of the game in

Game $\boxed{\mathrm{H}_0}$, H_1

$K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$; $\sigma' \leftarrow \varepsilon$
$C \leftarrow \emptyset$; $T \leftarrow \emptyset$
For $i = 1, \ldots, q$ do
  $j \leftarrow 0$; $\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{false}$
  $(m, \sigma') \leftarrow\$ \mathcal{M}(\sigma')$
  Repeat
    $j \leftarrow j + 1$; $r \leftarrow\$ \{0,1\}^{\mathsf{SE.rl}}$
    $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$
    $(v, t) \leftarrow\$ \{0,1\} \times [k]$
    If $(c \in C)$ then
      $\mathsf{bad} \leftarrow \mathsf{true}$
      $\boxed{(v, t) \leftarrow (v_{c'}, t_{c'})}$
    $(v_c, t_c) \leftarrow (v, t)$
    $C \leftarrow C \cup \{c\}$
    $\mathsf{success} \leftarrow (K_{\mathsf{SE}}[t_c] = v_c)$
    $\mathsf{out\text{-}of\text{-}time} \leftarrow (j = s)$
  Until ($\mathsf{success}$ OR $\mathsf{out\text{-}of\text{-}time}$)
  If $\mathsf{success}$ then $T \leftarrow T \cup \{t\}$
  Else $T \leftarrow T \setminus \{t\}$
Return $(T \neq [k])$

Adversary $\mathscr{F}$

$K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$; $\sigma' \leftarrow \varepsilon$
$T \leftarrow \emptyset$
For $i = 1, \ldots, q$ do
  $j \leftarrow 0$; $\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{false}$
  $(m, \sigma') \leftarrow\$ \mathcal{M}(\sigma')$
  Repeat
    $j \leftarrow j + 1$; $r \leftarrow\$ \{0,1\}^{\mathsf{SE.rl}}$
    $c \leftarrow \mathsf{SE.Enc}(K_{\mathsf{SE}}, m; r)$
    $(v_c, t_c) \leftarrow\$ \mathrm{FN}(c)$
    $\mathsf{success} \leftarrow (K_{\mathsf{SE}}[t_c] = v_c)$
    $\mathsf{out\text{-}of\text{-}time} \leftarrow (j = s)$
  Until ($\mathsf{success}$ OR $\mathsf{out\text{-}of\text{-}time}$)
  If $\mathsf{success}$ then $T \leftarrow T \cup \{t\}$
  Else $T \leftarrow T \setminus \{t\}$
If $(T \neq [k])$ then return 1
Else return 0

Game H_4

$K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$; $T \leftarrow \emptyset$
For $i = 1, \ldots, q$ do
  Repeat
    $(v, t) \leftarrow\$ \{0,1\} \times [k]$
  Until $(K_{\mathsf{SE}}[t] = v)$
  $T \leftarrow T \cup \{t\}$
Return $(T \neq [k])$

Game $\boxed{\mathrm{H}_2}$, H_3

$K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$; $T \leftarrow \emptyset$
For $i = 1, \ldots, q$ do
  $j \leftarrow 0$; $\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{false}$
  Repeat
    $j \leftarrow j + 1$
    $(v, t) \leftarrow\$ \{0,1\} \times [k]$
    $\mathsf{success} \leftarrow (K_{\mathsf{SE}}[t] = v)$
    If ($j = s$ AND $\mathsf{success} = \mathsf{false}$) then
      $\mathsf{bad} \leftarrow \mathsf{true}$
      $\boxed{\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{true}}$
  Until ($\mathsf{success}$ OR $\mathsf{out\text{-}of\text{-}time}$)
  If $\mathsf{success}$ then $T \leftarrow T \cup \{t\}$
  Else $T \leftarrow T \setminus \{t\}$
Return $(T \neq [k])$

Game H_5

$K_{\mathsf{SE}} \leftarrow\$ \{0,1\}^{\mathsf{SE.kl}}$; $T \leftarrow \emptyset$
For $i = 1, \ldots, q$ do
  $t \leftarrow\$ [k]$; $T \leftarrow T \cup \{t\}$
For $\ell = 1, \ldots, k$ do $\mathsf{bad}_\ell \leftarrow (\ell \notin T)$
$\mathsf{bad} \leftarrow (\mathsf{bad}_1 \vee \cdots \vee \mathsf{bad}_k)$
Return bad

Figure 4: Games and adversary for proof of Theorem 4.1.

which the adversary executes, so that the time of oracle calls is included. Thus for example the reason the time for $\mathcal{M}$ does not show up in the time of $\mathscr{F}$ below is that KR runs $\mathcal{M}$ already.

Theorem 4.1 *Let* SE *be a symmetric encryption scheme and let* $k = \mathsf{SE.kl}$. *Let* $\mathsf{F} : \{0,1\}^{\mathsf{F.kl}} \times \{0,1\}^* \to \{0,1\} \times [k]$ *be a PRF. Let* $q, s \geq 1$ *and let* A *be defined as in Fig. 3. Let* $\mathcal{M}$ *be an arbitrary message distribution. Then we can build PRF adversary* $\mathscr{F}$ *such that*

$$\mathbf{Adv}^{\mathrm{kr}}_{\mathsf{SE},\mathcal{M}}(\mathsf{A}) \geq 1 - \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{F}}(\mathscr{F}) - \epsilon(q, s, k) \tag{2}$$

where

$$\epsilon(q, s, k) \leq ke^{-q/k} + q2^{-s} + q^2 s^2 \cdot 2^{-\mathbf{H}_\infty(\mathsf{SE})-1} \ . \tag{3}$$

The running time of $\mathscr{F}$ *is about the sum of the running times of* $\mathsf{A.Enc}$ *and* $\mathsf{A.Ext}$, *and it makes at most* qs *oracle queries.*

For example, the key for SE is typically an AES key so that $k = \mathsf{SE.kl} = 128$. We could let $q = 128 \cdot 7 = 896$ and $s = 13$, which implies $ke^{-q/k} + q2^{-s} \leq 1/4$. We can assume $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{F}}(\mathscr{F})$ is negligible. Then as long as $\mathbf{H}_\infty(\mathsf{SE}) \geq 28$, our attack has advantage around $1/2$. The q, s values are quite small, making the attack quite practical.

That $\mathcal{M}$ may be *any* distribution makes the ASA particularly strong. The distribution $\mathcal{M}$ represents the choice of messages made by the encryptor. We are saying the attack works regardless of these choices. It is thus a known message attack, not a chosen message attack.

Proof of of Theorem 4.1: Game H_0 of Fig. 4 includes the boxed code while H_1 does not. Game H_0 implements game KR of Fig. 2 with A being as defined in Fig. 3, with two changes: (1) $\mathsf{F}(K_{\mathsf{A}}, \cdot)$ is replaced by a lazily sampled random function, and (2) the game returns true when key recovery fails rather than when it succeeds. Let prf adversary $\mathscr{F}$ be as defined in the second panel of Fig. 4. It uses its $\textsc{Fn}$ oracle where A would use $\mathsf{F}(K_{\mathsf{A}}, \cdot)$. It returns 1 when key recovery fails and 0 when it succeeds. Letting b_{prf} denote the challenge bit in game PRF, we have

$$\begin{aligned} \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \mid b_{\mathrm{prf}} = 1] &\geq 1 - \mathbf{Adv}_{\mathsf{SE}}^{\mathrm{kr}}(\mathsf{A}) \\ 1 - \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \mid b_{\mathrm{prf}} = 0] &= \Pr[\mathrm{H}_0] \ . \end{aligned}$$

The reason the first equation above is an inequality rather than an equality is that $\mathsf{A.Ext}$ initially sets all bits of K to 0 and, so, for a particular bit, it may by chance end up having the right value even when that bit is not set in the For loop. Now, using Eq. (1), we have

$$\begin{aligned} &\mathbf{Adv}_{\mathsf{F}}^{\mathrm{prf}}(\mathscr{F}) \\ &= \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \mid b_{\mathrm{prf}} = 1] - (1 - \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \mid b_{\mathrm{prf}} = 0]) \\ &\geq 1 - \mathbf{Adv}_{\mathsf{SE}}^{\mathrm{kr}}(\mathsf{A}) - \Pr[\mathrm{H}_0] \ . \end{aligned}$$

Re-arranging terms, we have

$$\mathbf{Adv}_{\mathsf{SE}}^{\mathrm{kr}}(\mathsf{A}) \geq 1 - \mathbf{Adv}_{\mathsf{F}}^{\mathrm{prf}}(\mathscr{F}) - \Pr[\mathrm{H}_0] \ .$$

We let

$$\epsilon(q, s, k) = \Pr[\mathrm{H}_0] \ .$$

This establishes Eq. (2). We now proceed to upper bound $\epsilon(q, s, k)$. Since games $\mathrm{H}_0, \mathrm{H}_1$ are identical until bad, the fundamental lemma of game playing [4] says that

$$\epsilon(q, s, k) = \Pr[\mathrm{H}_0] \leq \Pr[\mathrm{H}_1] + \Pr[\mathrm{H}_1 \text{ sets } \mathsf{bad}] \ .$$

Game H_1 sets bad when there is a collision in the ciphertexts. Since at most qs ciphertexts are created we have

$$\begin{aligned} \Pr[\mathrm{H}_1 \text{ sets } \mathsf{bad}] &\leq \binom{qs}{2} \cdot 2^{-\mathbf{H}_\infty(\mathsf{SE})} \\ &\leq q^2 s^2 \cdot 2^{-\mathbf{H}_\infty(\mathsf{SE})-1} \ . \end{aligned}$$

We now proceed to upper bound $\Pr[\mathrm{H}_1]$. The ciphertexts being chosen in this game are not relevant to its outcome, since (v_c, t_c) is chosen at random each time. Furthermore, the setting of the $\mathsf{out\text{-}of\text{-}time}$ flag only has an effect on the behavior of the game if the $\mathsf{success}$ flag is not already set at the time. This leads to game H_2 of Fig. 4, which includes the boxed code. We have

$$\Pr[\mathrm{H}_1] = \Pr[\mathrm{H}_2] \leq \Pr[\mathrm{H}_3] + \Pr[\mathrm{H}_3 \text{ sets } \mathsf{bad}] \ ,$$

the inequality by the fundamental lemma of game playing [4] because games $\mathrm{H}_2, \mathrm{H}_3$ are identical until bad. In game H_3 the boxed code is not included, with the result that the repeat loop continues until success. Since each iteration is successful with probability 1/2 we have

$$\Pr[\mathrm{H}_3 \text{ sets } \mathsf{bad}] \leq q 2^{-s} \ .$$

Game I_0	Adversary $\mathscr{F}$	**Game** I_1, $\boxed{\mathrm{I}_2}$

$b \leftarrow_\$ \{0,1\}$; $C \leftarrow \emptyset$
$b' \leftarrow_\$ \mathscr{D}^{\textsc{Enc}}$; Return $(b = b')$
$\underline{\textsc{Enc}(K, m)}$
If $(b = 1)$ then
  $c \leftarrow_\$ \mathsf{SE.Enc}(K, m)$
Else
  $j \leftarrow 0$; $\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{false}$
  Repeat
  $j \leftarrow j + 1$; $r \leftarrow_\$ \{0,1\}^{\mathsf{SE.rl}}$
  $c \leftarrow \mathsf{SE.Enc}(K, m; r)$
  $(v, t) \leftarrow_\$ \{0,1\} \times [k]$
  If $(c \in C)$ then
  $(v, t) \leftarrow (v_{c'}, t_{c'})$
  $(v_c, t_c) \leftarrow (v, t)$
  $C \leftarrow C \cup \{c\}$
  $\mathsf{success} \leftarrow (K[t_c] = v_c)$
  $\mathsf{out\text{-}of\text{-}time} \leftarrow (j = s)$
  Until ($\mathsf{success}$ OR $\mathsf{out\text{-}of\text{-}time}$)
Return (c, ϵ)
	$b \leftarrow_\$ \{0,1\}$
	$b' \leftarrow_\$ \mathscr{D}^{\textsc{Enc}}$
	If $(b = b')$ then return 1
	Else return 0
	$\underline{\textsc{Enc}(K, m)}$
	If $(b = 1)$ then
	  $c \leftarrow_\$ \mathsf{SE.Enc}(K, m)$
	Else
	  $j \leftarrow 0$; $\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{false}$
	  Repeat
	  $j \leftarrow j + 1$; $r \leftarrow_\$ \{0,1\}^{\mathsf{SE.rl}}$
	  $c \leftarrow \mathsf{SE.Enc}(K, m; r)$
	  $(v_c, t_c) \leftarrow_\$ \textsc{Fn}(c)$
	  $\mathsf{success} \leftarrow (K[t_c] = v_c)$
	  $\mathsf{out\text{-}of\text{-}time} \leftarrow (j = s)$
	  Until ($\mathsf{success}$ OR $\mathsf{out\text{-}of\text{-}time}$)
	Return (c, ϵ)
		$C \leftarrow \emptyset$
		$b' \leftarrow_\$ \mathscr{D}^{\textsc{Enc}}$; Return $(b' = 1)$
		$\underline{\textsc{Enc}(K, m)}$
		$j \leftarrow 0$;
		$\mathsf{out\text{-}of\text{-}time} \leftarrow \mathsf{false}$
		Repeat
		  $j \leftarrow j + 1$; $r \leftarrow_\$ \{0,1\}^{\mathsf{SE.rl}}$
		  $c \leftarrow \mathsf{SE.Enc}(K, m; r)$
		  $(v, t) \leftarrow_\$ \{0,1\} \times [k]$
		  If $(c \in C)$ then
		  $\mathsf{bad} \leftarrow \mathsf{true}$
		  $\boxed{(v, t) \leftarrow (v_{c'}, t_{c'})}$
		  $(v_c, t_c) \leftarrow (v, t)$
		  $C \leftarrow C \cup \{c\}$
		  $\mathsf{success} \leftarrow (K[t_c] = v_c)$
		  $\mathsf{out\text{-}of\text{-}time} \leftarrow (j = s)$
		Until ($\mathsf{success}$ OR $\mathsf{out\text{-}of\text{-}time}$)
		Return (c, ϵ)

Figure 5: Games an adversary for proof of Theorem 4.2.

We proceed to upper bound $\Pr[\mathrm{H}_3]$. Since the sampling in H_3 continues until success we have

$$\Pr[\mathrm{H}_3] = \Pr[\mathrm{H}_4] = \Pr[\mathrm{H}_5] \ .$$

We proceed to upper bound $\Pr[\mathrm{H}_5]$. Let $p_\ell = \Pr[\mathrm{H}_5 \text{ sets } \mathsf{bad}_\ell]$ for $\ell \in [k]$. Now we can use a standard coupon collector problem analysis. For any particular $\ell \in [k]$ we have

$$p_\ell = \left(1 - \frac{1}{k}\right)^q \leq e^{-q/k} \ .$$

Thus

$$\Pr[\mathrm{H}_5] \leq p_1 + \cdots + p_k \leq k e^{-q/k} \ .$$

Putting the inequalities together yields Eq. (3). ∎

<u>Strong undetectability.</u> The following theorem says that the ASA of our adversary A of Fig. 3 is strongly undetectable.

Theorem 4.2 *Let* SE *be a symmetric encryption scheme and let* $k = \mathsf{SE.kl}$. *Let* $\mathsf{F} : \{0,1\}^{\mathsf{F.kl}} \times \{0,1\}^* \to \{0,1\} \times [k]$ *be a PRF. Let* $q, s \geq 1$ *and let* A *be defined as in Fig. 3. Let* $\mathscr{D}$ *be an adversary against the strong undetectability of* A *that makes at most* n *queries to its* $\textsc{Enc}$ *oracle. Then we can build PRF adversary* $\mathscr{F}$ *such that*

$$\mathbf{Adv}^{\mathrm{sdet}}_{\mathsf{SE},\mathsf{A}}(\mathscr{D}) \leq 2\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{F}}(\mathscr{F}) + n^2 s^2 \cdot 2^{-\mathbf{H}_\infty(\mathsf{SE})}. \tag{4}$$

The running time of $\mathscr{F}$ *is about that of* $\mathscr{D}$ *and it makes at most* ns *oracle queries.*

Proof of of Theorem 4.2: Game I_0 of Fig. 5 implements SDET with $\mathsf{F}(K_\mathsf{A}, \cdot)$ replaced by a

lazily sampled random function. Let prf adversary $\mathscr{F}$ be as defined in the second panel of Fig. 5. It runs $\mathscr{D}$, itself simulating the latter's Enc oracle, in the process appealing to its own Fn oracle. Letting $b_{\rm prf}$ denote the challenge bit in game PRF, we have

$$\begin{aligned} \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \,|\, b_{\rm prf} = 1] &= \frac{1}{2} + \frac{1}{2}\mathbf{Adv}_{\mathsf{SE,A}}^{\rm sdet}(\mathscr{D}) \\ 1 - \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \,|\, b_{\rm prf} = 0] &= \Pr[\mathrm{I}_0] \ . \end{aligned}$$

Thus, using Eq. (1), we have

$$\begin{aligned} &\mathbf{Adv}_{\mathsf{F}}^{\rm prf}(\mathscr{F}) \\ &= \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \,|\, b_{\rm prf} = 1] - (1 - \Pr[\mathrm{PRF}_{\mathsf{F}}^{\mathscr{F}} \,|\, b_{\rm prf} = 0]) \\ &= \frac{1}{2} + \frac{1}{2}\mathbf{Adv}_{\mathsf{SE,A}}^{\rm sdet}(\mathscr{D}) - \Pr[\mathrm{I}_0] \ . \end{aligned}$$

Re-arranging terms, we have

$$\mathbf{Adv}_{\mathsf{SE,A}}^{\rm sdet}(\mathscr{D}) \le 2\mathbf{Adv}_{\mathsf{F}}^{\rm prf}(\mathscr{F}) + (2\Pr[\mathrm{I}_0] - 1) \ .$$

We proceed to upper bound $2\Pr[\mathrm{I}_0] - 1$. Consider games $\mathrm{I}_1, \mathrm{I}_2$ of Fig. 5. Game I_2 includes the boxed code while I_1 does not. In both, (v_c, t_c) is picked at random. However, in I_2, if it turns out that c was already seen, then the game corrects, resetting (v_c, t_c) to its prior value. As a result, I_2 is equivalent to I_0 with the challenge bit b in the latter set to $b = 0$, except it returns true when $b' = 1$ instead of $b' = 0$. On the other hand, in I_1, the choice of (v_c, t_c) is independent of the choice of c and does not impact the distribution of the latter, making it equivalent to I_0 with $b = 1$. Thus we have

$$\begin{aligned} \Pr[\mathrm{I}_1] &= \Pr[\mathrm{I}_0 \,|\, b = 1] \\ \Pr[\mathrm{I}_2] &= 1 - \Pr[\mathrm{I}_0 \,|\, b = 0] \ . \end{aligned}$$

Additionally, games $\mathrm{I}_0, \mathrm{I}_1$ are identical until bad. Via the fundamental lemma of game playing [4] we have

$$\begin{aligned} 2\Pr[\mathrm{I}_0] - 1 &= \Pr[\mathrm{I}_0 \,|\, b = 1] - (1 - \Pr[\mathrm{I}_0 \,|\, b = 0]) \\ &= \Pr[\mathrm{I}_1] - \Pr[\mathrm{I}_2] \\ &\le \Pr[\mathrm{I}_1 \text{ sets } \mathsf{bad}] \ . \end{aligned}$$

Game I_1 sets bad whenever the same value of c is returned twice by $\mathsf{SE.Enc}$. There are ns queries to Enc. Thus

$$\begin{aligned} \Pr[\mathrm{I}_1 \text{ sets } \mathsf{bad}] &\le \binom{ns}{2} \cdot 2^{-\mathbf{H}_\infty(\mathsf{SE})} \\ &\le n^2 s^2 \cdot 2^{-\mathbf{H}_\infty(\mathsf{SE})-1} \ . \end{aligned}$$

Putting everything together gives the desired Eq. (4). ▮

<u>Discussion.</u> We do not expect that most users will mount involved detection efforts. But the weakness of the ASA of BPR [3] is that is is relatively easily detectable, even without much effort, due to its use of state. The most obvious way for this to happen is that the state is reset, for example due to a system reboot. When this happens, the decrypter will be able to detect the subversion. This, however, is not captured by their notion of detectability. Strong undetectability

fills the gap. Our new ASA, achieving this and in particular being stateless, is harder to detect and thus more dangerous.

In a world of subversion, there are no panaceas. The extent to which big brother will risk detection in order to successfully recover a key is not known, but it would be natural that, all else being equal, big brother will pick the ASA that minimizes the chance of detection. This means an ASA like ours. Our work is intended to increase awareness and spur the search for deterrents.

5 Defenses

BPR [3] present schemes that are subversion resilient. They define unique ciphertext schemes, show they are subversion resilient, and then provide several ways to build them. These schemes are deterministic and stateful, meaning the encryptor and decryptor have to maintain a synchronized state.

Having strengthened the BPR undetectability condition to strong undetectability, a natural question is whether subversion resilience can still be achieved. In fact this strengthening makes no difference, and the schemes of BPR continue to achieve subversion resilience. This is because a subversion resilient scheme according to BPR [3] is one where any subversion satisfying the decryptability condition cannot succeed in their subversion game. Decryptability means that ciphertexts created by the subverted encryption algorithm decrypt properly under the real decryption algorithm, and it can be seen as represented a particular and fixed form of undetectability.

DFP [5] critique the BPR decryptability condition and instead suggest an alternative formulation of detectability represented by their $\overline{\text{DETECT}}$ game in which the subversion and detection adversaries run together, the latter getting a transcript of the interaction of the former with its oracles. They show that unique ciphertext schemes continue to achieve their notion. In this context, consideration of strong undetectability would involve modifying their game to a new one, $\overline{\text{SDETECT}}$. For example one might return the state to the detection adversary as part of the transcript, and also add a reset oracle Reset to allow state reset. Then the question is whether unique ciphertext schemes continue to meet this new definition. They do.

6 Acknowledgements

We thank Kaspar-David Buss for pointing out an error in the proof of Theorem 4.2. The April 2017 update of this paper corrects that error.

References

[1] G. Ateniese, B. Magri, and D. Venturi. Subversion-resilient signature schemes. Cryptology ePrint Archive, Report 2015/517, 2015. `http://eprint.iacr.org/2015/517`.

[2] M. Bellare, J. Kilian, and P. Rogaway. The security of cipher block chaining. In Y. Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 341–358. Springer, Aug. 1994.

[3] M. Bellare, K. G. Paterson, and P. Rogaway. Security of symmetric encryption against mass surveillance. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 1–19. Springer, Aug. 2014.

[4] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006.

[5] J. P. Degabriele, P. Farshim, and B. Poettering. A more cautious approach to security against mass surveillance. In G. Leander, editor, *FSE 2015*, volume 9054 of *LNCS*. Springer, Mar. 2015.

[6] Y. Dodis, C. Ganesh, A. Golovnev, A. Juels, and T. Ristenpart. A formal treatment of backdoored pseudorandom generators. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 101–126. Springer, Apr. 2015.

[7] E.-J. Goh, D. Boneh, B. Pinkas, and P. Golle. The design and implementation of protocol-based hidden key recovery. In C. Boyd and W. Mao, editors, *ISC 2003*, volume 2851 of *LNCS*, pages 165–179. Springer, Oct. 2003.

[8] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, Oct. 1986.

[9] I. Mironov and N. Stephens-Davidowitz. Cryptographic reverse firewalls. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 657–686. Springer, Apr. 2015.

[10] J. Patarin and L. Goubin. Asymmetric cryptography with S-boxes. In Y. Han, T. Okamoto, and S. Qing, editors, *ICICS 97*, volume 1334 of *LNCS*, pages 369–380. Springer, Nov. 1997.

[11] K. G. Paterson. Imprimitive permutation groups and trapdoors in iterated block ciphers. In L. R. Knudsen, editor, *FSE'99*, volume 1636 of *LNCS*, pages 201–214. Springer, Mar. 1999.

[12] V. Rijmen and B. Preneel. A family of trapdoor ciphers. In E. Biham, editor, *FSE'97*, volume 1267 of *LNCS*, pages 139–148. Springer, Jan. 1997.

[13] A. Russell, Q. Tang, M. Yung, and H.-S. Zhou. Cliptography: Clipping the power of kleptographic attacks. Cryptology ePrint Archive, Report 2015/695, 2015. `http://eprint.iacr.org/`.

[14] B. Schneier, M. Fredrikson, T. Kohno, and T. Ristenpart. Surreptitiously weakening cryptographic systems. Cryptology ePrint Archive, Report 2015/097, 2015. `http://eprint.iacr.org/2015/097`.

[15] G. J. Simmons. The prisoners' problem and the subliminal channel. In D. Chaum, editor, *CRYPTO'83*, pages 51–67. Plenum Press, New York, USA, 1983.

[16] G. J. Simmons. The subliminal channel and digital signature. In T. Beth, N. Cot, and I. Ingemarsson, editors, *EUROCRYPT'84*, volume 209 of *LNCS*, pages 364–378. Springer, Apr. 1985.

[17] G. J. Simmons. A secure subliminal channel (?). In H. C. Williams, editor, *CRYPTO'85*, volume 218 of *LNCS*, pages 33–41. Springer, Aug. 1986.

[18] G. J. Simmons. Subliminal communication is easy using the DSA. In T. Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 218–232. Springer, May 1994.

[19] A. Young and M. Yung. The dark side of "black-box" cryptography, or: Should we trust capstone? In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 89–103. Springer, Aug. 1996.

[20] A. Young and M. Yung. Kleptography: Using cryptography against cryptography. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 62–74. Springer, May 1997.

[21] A. Young and M. Yung. Monkey: Black-Box symmetric ciphers designed for MONopolizing KEYs. In S. Vaudenay, editor, *FSE'98*, volume 1372 of *LNCS*, pages 122–133. Springer, Mar. 1998.

[22] A. Young and M. Yung. A subliminal channel in secret block ciphers. In H. Handschuh and A. Hasan, editors, *SAC 2004*, volume 3357 of *LNCS*, pages 198–211. Springer, Aug. 2004.

[23] A. L. Young and M. Yung. Backdoor attacks on black-box ciphers exploiting low-entropy plaintexts. In R. Safavi-Naini and J. Seberry, editors, *ACISP 03*, volume 2727 of *LNCS*, pages 297–311. Springer, July 2003.