Security of Symmetric Encryption against Mass Surveillance

Mihir Bellare[1], Kenneth G. Paterson[2], and Phillip Rogaway[3]

[1] Dept. of Computer Science and Engineering, University of California San Diego, USA. cseweb.ucsd.edu/~mihir
[2] Information Security Group, Royal Holloway, University of London, UK. www.isg.rhul.ac.uk/~kp
[3] Dept. of Computer Science, University of California Davis, USA. www.cs.ucdavis.edu/~rogaway

Abstract. Motivated by revelations concerning population-wide surveillance of encrypted communications, we formalize and investigate the resistance of symmetric encryption schemes to mass surveillance. The focus is on algorithm-substitution attacks (ASAs), where a subverted encryption algorithm replaces the real one. We assume that the goal of "big brother" is undetectable subversion, meaning that ciphertexts produced by the subverted encryption algorithm should reveal plaintexts to big brother yet be indistinguishable to users from those produced by the real encryption scheme. We formalize security notions to capture this goal and then offer both attacks and defenses. In the first category we show that successful (from the point of view of big brother) ASAs may be mounted on a large class of common symmetric encryption schemes. In the second category we show how to design symmetric encryption schemes that avoid such attacks and meet our notion of security. The lesson that emerges is the danger of choice: randomized, stateless schemes are subject to attack while deterministic, stateful ones are not.

Table of Contents

1 Introduction

Overview. This paper is about the troubling possibility of mass surveillance by *algorithm-substitution attack* (ASA). Suppose that encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is to be implemented in closed-source software—think, for example, of implementing the CBC-AES encryption underlying the TLS record layer within Microsoft's Internet Explorer or Apple's Safari browsers, or in corresponding server-side code. An ASA replaces the executable code for the desired encryption algorithm $\mathcal{E}$ with, for example, the code of an NSA-authored alternative $\widetilde{\mathcal{E}}$.

ASAs have been discussed before, under various names, in particular falling under the banner of *kleptography*. This prescient idea was developed by Young and Yung starting in the 1990s [43, 44]. While some cryptographers seem to have dismissed kleptography as far-fetched, recent revelations suggest this attitude to be naïve [3]. ASAs may well be going on today, possibly on a massive scale. In this light we aim to provide a *formal* and *practical* treatment of ASAs, with a focus on *symmetric encryption*, an attractive target for real-world attacks. Building on, yet going further than, prior work, we fully and formally define security goals. We then come at ASAs from both ends, showing on the one hand how successful (from the point of view of big brother) ASAs may be mounted on standard schemes, and showing on the other hand how to design schemes that provably resist them. Our findings surface what we call *the danger of choice*: the trend towards flexibility and open-ended choices in protocols, often present for vendor flexibility or political compromise, works against us with regard to protection against ASAs, which are best defeated by stateful, deterministic encryption that curtails randomness and choice.

Model and definitions. The real encryption algorithm $\mathcal{E}$ takes, as usual, user key K, message M, and associated data A. It returns a ciphertext C. The subverted algorithm $\widetilde{\mathcal{E}}$ that substitutes for $\mathcal{E}$ takes the same inputs but also an additional, big-brother key, $\widetilde{K}$. It also returns a ciphertext.

With no restrictions on $\widetilde{\mathcal{E}}$, there would appear to be no hope of security, for $\widetilde{\mathcal{E}}$ can fold K into the ciphertext, say encrypted under $\widetilde{K}$, and big brother can use $\widetilde{K}$ to recover K. However, such an attack would be detected by users, who would see that ciphertexts fail to decrypt normally. Big brother aims to achieve compromise without detection: subverted ciphertexts should look like real ones, yet enable recovery of K or M. ASAs, in this view, live in a tension between detectability and success, the former working to curtail the latter. We will formally define metrics of both detectability and success.

We will require that ciphertexts produced by $\widetilde{\mathcal{E}}$ decrypt normally under the decryption algorithm $\mathcal{D}$ of the base scheme. This decryptability condition is the most basic form of undetectability. But we expect that big brother will aim to evade more sophisticated forms of detection. We formalize detection security as requiring that real and subverted ciphertexts are indistinguishable even to a test that knows some users' keys but does not know $\widetilde{K}$.

Success refers to big brother's ability to obtain knowledge about user data from subverted ciphertexts. Certainly an ASA allowing big brother to recover the user key K from any ciphertext is successful, but for positive results (defeating big brother) we want more. We formalize surveillance security as the requirement that big brother, even with its key $\widetilde{K}$, cannot differentiate real ciphertexts from subverted ones.

The duality between detection and surveillance security is reflected in our formalizations. Both require indistinguishability of real and subverted ciphertexts to an adversary, the difference being that in detection the adversary knows the user keys but not the big-brother key, and in surveillance it's the other way around. We remark that, in both cases, our formalizations are multi-user, meaning there are many users (but a single subverter).

Mounting ASAs. We show that most symmetric encryption schemes succumb to damaging ASAs. Our attacks recover the user key K from subverted ciphertexts while remaining undetectable. These attacks apply to base schemes that are randomized and stateless. Building on [16], we first describe what

we call IV-replacement attacks, where the initial vector in a blockcipher mode of operation is used to communicate to big brother an encryption under $\widetilde{K}$ of the user key K. Then we describe a more general ASA that we call the biased-ciphertext attack. This makes few assumptions on the structure of the base scheme and succeeds by creating ciphertexts that are not distributed quite like real ones. They are biased in a way that reveals bits of the user key to a holder of $\widetilde{K}$, but we show that the bias is undetectable without knowledge of $\widetilde{K}$. The difficulty here is showing undetectability even for tests that know the user key K, and for the analysis we prove an information-theoretic lemma about biased functions. Beyond presenting generic attacks we discuss how encryption in SSL/TLS, IPsec, and SSH can be subverted by these means. The conclusion is that randomized, stateless schemes, including deployed ones, invariably fall to even generic ASAs.

Defeating ASAs. We aim to build symmetric encryption schemes that resist ASAs, meaning achieve surveillance security in the formal sense we define. Given the above, such schemes need to be stateful and deterministic. But not every such scheme works. The difficulty with provably achieving surveillance security is that standard security properties of the base scheme, such as its privacy or authenticity, are of no particular use towards the new goal. The reason is that these properties rely on the adversary not knowing the key K. But in the surveillance setting, the subverted ciphertexts are being created by an algorithm, $\widetilde{\mathcal{E}}$, that knows K, and can thus compromise privacy or authenticity to make subverted ciphertexts look different from real ones, and in a way useful to big brother. Nonetheless, we show that security is achievable by relying on combinatorial properties of the scheme. We define what it means for a base symmetric encryption scheme to have *unique ciphertexts* and then show that every unique-ciphertext scheme meeting the decryptability condition is secure against ASAs. This provides a strong anti-surveillance guarantee: no ASA will succeed in differentiating real from subverted ciphertexts, let alone recovering the message or a user's key. We show this assuming only minimal undetectability—decryptability, meaning that subverted ciphertexts must remain decryptable by the decryption algorithm of the base scheme.

To realize concrete benefits from this general result, we need to find unique-ciphertext symmetric encryption schemes. Here we give a simple construction based on a variable-input-length PRP. We present a more practical result, showing how any nonce-based symmetric encryption scheme [34, 35] may be transformed into a unique ciphertext stateful deterministic scheme while preserving efficiency. Using existing nonce-based encryption schemes like CCM, GCM, or OCB, this yields practical designs of surveillance-resistant symmetric encryption.

Asymmetric ASAs. For simplicity, our main definitions only capture the case in which big brother embeds a symmetric key $\widetilde{K}$ into subverted software. It is obviously useful to replace this with a public key, the corresponding secret key being held by big brother, so that reverse engineering of a subverted encryption algorithm will not confer the capabilities that big brother aims to keep to itself. The necessary definitional extensions, which are small, are described in Appendix B.

Scope. Our paper is deliberately of restricted scope: we consider ASAs only for symmetric encryption schemes. In reality, encryption schemes are deployed as part of larger cryptographic protocols and these protocols will afford additional opportunities for algorithmic subversion. To pick one example, a protocol might involve the transmission of a nonce for authentication purposes during a key-exchange phase. This nonce could be chosen so as to directly leak an ensuing session key. Or it could be chosen to leak the internal state of a back-doored PRNG, indirectly revealing future session keys. This technique has been posited as a subversion method for SSL/TLS [9].

Our scope also means that we exclude subversion attempts that exploit side-channels in implementations. For example, our model does not capture timing information, so attacks in which the encryption key is leaked through fine-grained timing behaviour of the encryption algorithm fall outside our notions.

Big brother's subverted $\widetilde{\mathcal{E}}$ could stutter the times at which ciphertexts or their blocks are produced; this might be sufficient to build a covert channel with adequate bandwidth to convey the session key. Such timing approaches have been used to infer information about user keystrokes over SSH connections [39].

The limitations on scope imply that our positive security results are certainly not definitive in terms of eliminating all subversion possibilities for a symmetric encryption scheme deployed within a real-world system. Still, a limited scope has merit. First, symmetric encryption is fundamental to secure communications, so it's important to study this primitive's susceptibility to subversion. Second, our model fits well within the scenario where an agency subverts encryption software, like a crypto library, rather than a particular protocol built on that library. Third, the positive results we provide, showing that ASAs on certain schemes are impossible, confine big brother to other avenues of attack, which may be less attractive. Finally, we aim to lay foundational results, in the modern, provable-security style, that can be built upon by succeeding researchers to broaden the scope of surveillance-resistant protocols to include tasks such as authenticated key exchange. It should eventually be possible to have a corpus of protocols, and even system-level code analysis, to provide strong guarantees on the ineffectiveness ASAs.

The danger of choice. The characteristic of modern encryption schemes that makes ASAs possible is the freedom-of-choice routinely provided by protocols, as well as the unverifiability of mandated randomness. Consider a symmetric encryption scheme that requires a user to select a 128-bit IV. The specification might say that the IV *should* be chosen uniformly at random, or it might even say that it *must* be so chosen. But, either way, the black-box behavior of the encryption scheme will never reveal if uniform random bits were used. Because of this, there is no way to ensure that the IV is not selected in a manner that will covertly communicate a session key to an agency engaged in mass surveillance—which we exploit in our IV-replacement attack. Similarly, if a scheme permits variable-length padding there will be no way to ensure that the amount of padding is not used as a covert channel to transmit a user's key.

The ultimate conclusion of this paper is that unverifiable algorithmic choice can be a significant liability. We have in some sense come full-circle. In their classical paper on probabilistic encryption [17], Goldwasser and Micali explained the danger of deterministic public-key encryption: leaking that one ciphertext is the repetition of another, or allowing a ciphertext to be decrypted by trial-encryption. But these threats can be eliminated without the use of probabilism—namely, through the use of state. For the most conventional setting in symmetric encryption—realizing a reliable, encrypted channel—ASAs provide one motivation for deterministic, stateful schemes, for sender and receiver both. We believe that there are further benefits to such schemes, including improved utility for software testing and the elimination of any need, post key-generation, to harvest unpredictable random bits.

Related work. Young and Yung have developed an extensive body of work on what they call *kleptography*, beginning with [43, 44]. This concerns the deliberate subversion of cryptosystems to provide backdoor capabilities; our work is a special case. While much of their work has focused on the public-key setting, Young and Yung have also considered attacks on protocols like Kerberos, and developed blockciphers containing backdoors for the black-box setting (ie, where the code of the blockcipher is not made available for inspection) [46, 48, 47]. In the light of recent revelations, we contend that kleptography deserves to play a larger role in the future development of our field. Additional work on back-doored blockciphers can be found in [32, 29, 30]. This entire line of work has focused on building schemes with deliberately-inserted and hard-to-detect backdoors. By contrast, we also provide positive results, constructing schemes that are provably hard to subvert.

Goh, Boneh, Pinkas and Golle [16] consider the problem of adding key recovery to the SSL/TLS and SSH protocols. Some passages of this 2003 paper now sound prophetic: *The government can convince major software vendors to distribute SSL/TLS or SSH2 implementations with hidden and unfilterable key recovery. ... Users will not notice the key recovery mechanism because the scheme is hidden.* [16, Section 2.2]. Goh et al. suggest that when the server needs a random nonce, it can use in its place an encryption

of the session key computed under the escrow key. We build on this idea to consider more general classes of attack on symmetric encryption schemes. Most of the work of Goh et al. is focused on practical aspects of enabling key recovery, including the development of a prototype system for SSL/TLS.

The problem of inserting backdoors and key-recovery defects into cryptographic schemes is closely related to the topic of *subliminal channels*, whose extensive literature begins with [38] and the study of covert channels [27]. There is a similarly extensive body of work on the exploitation, measurement, and elimination of timing side channels, both in cryptographic and non-cryptographic settings, with representative examples including [8, 24].

Further remarks. We posed our initial question in the context of closed-source software. However the sheer complexity of cryptographic libraries like OpenSSL, and the small number of experts who review such code, makes it plausible that ASAs might be carried out against open-source software. Note too that even when code appears to be "clean," there's always the possibility of code being subverted at compilation or run time, by subverting the compiler or interpreter [40]. And there's certainly the possibility of performing ASAs on hardware-based cryptography, a prospect rendered all the easier by the widespread use of countermeasures *intended* to shield algorithmic internals from inspection.

We do not know if ASAs are among the techniques used to make TLS-encrypted traffic available under warrantless surveillance [3]. We offer no empirical evidence in this direction. We hope that other researchers are seeking it out, which is necessary for understanding the actual nature of our communication infrastructure.

2 Preliminaries

Notation. A string means a member of $\{0,1\}^*$, and $\perp \notin \{0,1\}^*$ denotes a special symbol standing for "invalid" or "reject." If S is a set then $x \twoheadleftarrow S$ denotes sampling x uniformly at random from S.

Syntax. Our syntax for symmetric encryption encompasses encryption that is probabilistic, deterministic, or stateful; and decryption that is deterministic or stateful. We allow associated data (AD), in order that our basic syntax encompass this practically-important component of authenticated encryption.

A scheme for *symmetric encryption* is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The *key space* $\mathcal{K}$ is a finite nonempty set. The *encryption algorithm* $\mathcal{E}$ is a possibly randomized algorithm that maps a four-tuple of strings K, M, A, σ to a pair of strings $(C, \sigma') \twoheadleftarrow \mathcal{E}(K, M, A, \sigma)$. The arguments to $\mathcal{E}$ represent the key, message (plaintext), associated data and current state. The output consists of the ciphertext C and revised state σ'. The *decryption algorithm* $\mathcal{D}$ is a deterministic algorithm that maps a four-tuple of strings (K, C, A, σ) to a pair of strings $(M, \sigma') \leftarrow \mathcal{D}(K, C, A, \sigma)$.

Algorithms $\mathcal{E}$ and $\mathcal{D}$ are said to *reject* if they return a pair with first component of $\perp$, and to *accept* otherwise. We may write $\mathcal{E}_K(M, A, \sigma)$ and $\mathcal{D}_K(C, A, \sigma)$ for $\mathcal{E}(K, M, A, \sigma)$ and $\mathcal{D}(K, C, A, \sigma)$, respectively. We adopt the convention that $\mathcal{E}$ and $\mathcal{D}$ return $(\perp, \perp)$ if any argument is $\perp$. In addition, whether or not $C_i = \perp$ is allowed to depend only on $|M_1|$, $|A_1|, \ldots$, $|M_{i-1}|$, and $|A_{i-1}|$. This eliminates pointless degeneracies.

We say that $\mathcal{E}$ is stateless if the second component of any output of $\mathcal{E}$ on any inputs is ε, and likewise for $\mathcal{D}$. We say that Π is stateless if both $\mathcal{E}$ and $\mathcal{D}$ are stateless. In this case, we drop the second component of the output of both algorithms, so that $\mathcal{E}$ now returns just a ciphertext and $\mathcal{D}$ just a message. We also drop the last (state) input to $\mathcal{D}$ and, for $\mathcal{E}$, think of it as the coins of the algorithm, dropping which is regarded as having the coins being chosen at random. In this way, when Π is stateless, we recover the conventional syntax.

It is well understood that encryption must be stateful or probabilistic to achieve IND-CPA privacy and decryption must be stateful to avoid replay attacks. Our work will show that decryption must be stateful to avoid algorithm-substitution attacks.

CORRECTNESS. We say that $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is *correct*, or meets the correctness condition, if, when the sender encrypts a sequence of messages and the receiver decrypts the resulting sequence of ciphertexts in order, the receiver will get back what the sender started with. To be clear what this means in our current stateful context, we now proceed more formally. Saying that encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is correct means that for all q, all $M_1, \ldots, M_q \in \{0,1\}^*$ and all $A_1, \ldots, A_q \in \{0,1\}^*$, the following game returns true with probability zero:

$\sigma_0, \tau_0 \leftarrow \varepsilon$
For $i = 1, \ldots, q$ do $(C_i, \sigma_i) \twoheadleftarrow \mathcal{E}(K, M_i, A_i, \sigma_{i-1})$; $(M'_i, \tau_i) \leftarrow \mathcal{D}(K, C_i, A_i, \tau_{i-1})$
Return $((\forall i \,:\, C_i \neq \perp)$ and $(\exists i \,:\, M_i \neq M'_i))$

We will only consider schemes that are correct in this sense.

SECURITY NOTIONS. We recall a standard notion of privacy for symmetric encryption [4, 5, 34]. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme and let $\mathscr{A}$ be an adversary. Consider the following game:

Game $\mathrm{PRIV}^{\mathscr{A}}_{\Pi}$	$\underline{\textsc{Enc}(M, A)}$
$K \twoheadleftarrow \mathcal{K}$; $\sigma \leftarrow \varepsilon$; $b \twoheadleftarrow \{0,1\}$	If $b = 1$ then $(C, \sigma) \twoheadleftarrow \mathcal{E}(K, M, A, \sigma)$
$b' \leftarrow \mathscr{A}^{\textsc{Enc}}$; Return $(b = b')$	Else $(C, \sigma) \twoheadleftarrow \mathcal{E}(K, 0^{\lvert M\rvert}, A, \sigma)$
	Return C

Let $\mathbf{Adv}^{\mathrm{priv}}_{\Pi}(\mathscr{A}) = 2\Pr[\mathrm{PRIV}^{\mathscr{A}}_{\Pi} \Rightarrow \mathsf{true}] - 1$ be the privacy advantage of adversary $\mathscr{A}$. Positive results will provide schemes secure in this sense and also resistant to surveillance as we will define in Section 3.

3 Subverting Encryption

We now ask what it would mean for a symmetric encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ to fall to an algorithm substitution attack (ASA). An attacker $\mathscr{B}$ (for "big brother") wants to subvert an encryption scheme *en masse*. We assume it is able to arrange that subverted encryption code $\widetilde{\mathcal{E}}_{\widetilde{K}}$ is used in place of $\mathcal{E}$. (The subscript indicates that a key $\widetilde{K}$ chosen by $\mathscr{B}$ may be embedded in the code.) $\mathscr{B}$ wants its subversion to be successful and yet undetected. The former means that from observing only ciphertexts computed under the subverted algorithm, $\mathscr{B}$ can compromise privacy. (For example, it can, using $\widetilde{K}$, efficiently recover the plaintexts underlying the ciphertexts.) This captures the relevant attack scenario where $\mathscr{B}$ is able, through mass surveillance of network traffic, to intercept bulk ciphertexts at will. The latter means that the subverted encryption algorithm should produce ciphertexts that look alright. The most basic form of the latter requirement is that they correctly decrypt under the decryption algorithm $\mathcal{D}$ of the base scheme, but we expect that big brother would prefer to evade even more sophisticated attempts at detection.

One can consider subverting an encryption scheme's privacy, authenticity, or both. One can also consider subversion for public-key schemes or for other cryptographic goals, like key exchange. There are possibilities for algorithm-substitution attacks (ASAs) in all these settings. Here we limit the scope to subversion aimed at compromising the privacy of a symmetric encryption scheme. The extensions to cover additional schemes is an obvious and important target for future research.

SUBVERSIONS. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. A *subversion* of Π is a triple $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$. The *master-key space* $\widetilde{\mathcal{K}}$ is a finite nonempty set. The *subverted encryption algorithm* $\widetilde{\mathcal{E}}$ is a (possibly randomized) algorithm that maps a six-tuple of strings $(\widetilde{K}, K, M, A, \sigma, i)$ to a pair of strings (C, σ'). Here σ and σ' are the current and updated states, respectively, indicating that $\widetilde{\mathcal{E}}$ may be stateful. The input i represents some public information identifying a user encrypting under K and is assumed

Game $\text{DETECT}^{\mathscr{U}}_{\Pi,\widetilde{\Pi}}$	**Game** $\text{SURV}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}}$
$b \twoheadleftarrow \{0,1\}$; $\widetilde{K} \twoheadleftarrow \widetilde{\mathcal{K}}$; $b' \twoheadleftarrow \mathscr{U}^{\text{KEY},\text{ENC}}$	$b \twoheadleftarrow \{0,1\}$; $\widetilde{K} \twoheadleftarrow \widetilde{\mathcal{K}}$; $b' \twoheadleftarrow \mathscr{B}^{\text{KEY},\text{ENC}}(\widetilde{K})$
Return $(b = b')$	Return $(b = b')$
KEY(i)	KEY(i)
If $(K_i = \bot)$ then $K_i \twoheadleftarrow \mathcal{K}$; $\sigma_i \leftarrow \varepsilon$	If $(K_i = \bot)$ then $K_i \twoheadleftarrow \mathcal{K}$; $\sigma_i \leftarrow \varepsilon$
Return K_i	Return ε
ENC (M, A, i)	ENC (M, A, i)
If $(K_i = \bot)$ then return $\bot$	If $(K_i = \bot)$ then return $\bot$
If $(b = 1)$ then $(C, \sigma_i) \twoheadleftarrow \mathcal{E}(K_i, M, A, \sigma_i)$	If $(b=1)$ then $(C, \sigma_i) \twoheadleftarrow \mathcal{E}(K_i, M, A, \sigma_i)$
Else $(C, \sigma_i) \twoheadleftarrow \widetilde{\mathcal{E}}(\widetilde{K}, K_i, M, A, \sigma_i, i)$	Else $(C, \sigma_i) \twoheadleftarrow \widetilde{\mathcal{E}}(\widetilde{K}, K_i, M, A, \sigma_i, i)$
Return C	Return C

Fig. 1. Games used to define detection and surveillance security of subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ of encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$.

different for all keys. Such information is usually available in a system, perhaps a MAC address or an IP address, and we allow $\widetilde{\mathcal{E}}$ to take it as input because we cannot realistically disallow a subverter from having or using such information.

The *plaintext-recovery algorithm* $\widetilde{\mathcal{D}}$ takes $\widetilde{K}, \boldsymbol{C}, \boldsymbol{A}, i$ where $\boldsymbol{C}$ is a vector of ciphertexts, $\boldsymbol{A}$ is a vector of associated data and i is again the identity associated to the key K whose usage is being subverted. The algorithm attempts to produce a vector of corresponding plaintexts $\boldsymbol{M}$. How effectively it does this will vary. For example, the plaintext-recovery algorithm $\widetilde{\mathcal{D}}$ may *always* find the plaintext, for every ciphertext in the list, regardless of the length of the list. Or it may effectively perform a key recovery attack first, then simply decrypt the ciphertexts, but require many ciphertexts. In describing the severity of a practical ASA, we will explicitly specify $\widetilde{\mathcal{D}}$ and quantify how good a job it does—a break that always finds the plaintext, or something else. For defining our security notion, however, we will ignore $\widetilde{\mathcal{D}}$, for the very strong notion we shall give implies the inexistence of any practical plaintext-recovery algorithm $\widetilde{\mathcal{D}}$.

DECRYPTABILITY. We say that $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ satisfies the *decryptability condition* relative to $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ if $(\widetilde{\mathcal{K}} \times \mathcal{K}, \widetilde{\mathcal{E}}, \mathcal{D}')$ is a correct encryption scheme where $\mathcal{D}'$ is defined by $\mathcal{D}'((\widetilde{K}, K), C, A, \sigma) = \mathcal{D}(K, C, A, \sigma)$. Thus, although algorithm $\widetilde{\mathcal{E}}$ operates on a key $(\widetilde{K}, K)$ different from the key K of the base scheme Π, a party possessing only K can decrypt $\widetilde{\mathcal{E}}$-encrypted plaintexts using the legitimate decryption algorithm $\mathcal{D}$. This represents the most basic form of resistance to detection, and we will assume any subversion must meet it.

DETECTION ADVANTAGE. By detectability, we refer to the ability of ordinary users—they know their secret keys, but not the master key—to tell, from the ciphertexts, if encryption is happening by the real or subverted algorithm. In the absence of any detectability condition, subversion is always possible. The decryptability condition we gave above embodies a particularly basic form of detection, in that failure to meet this condition is likely to lead to detection. However, we expect that big brother wants to evade not just this, but more sophisticated forms of detection. We now define what it means to do so. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ be a subversion of it. Let $\mathscr{U}$ be an algorithm representing a detection test being run by users. Let

$$\mathbf{Adv}^{\text{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) \;=\; 2\Pr[\text{DETECT}^{\mathscr{U}}_{\Pi,\widetilde{\Pi}} \Rightarrow \mathsf{true}] - 1$$

where game DETECT is shown on the left of Fig. 1. This measures the ability of test $\mathscr{U}$ to detect an ASA. In this game, $\mathscr{U}$ must detect whether it receives ciphertexts produced by $\mathcal{E}$ or by $\widetilde{\mathcal{E}}$. Via oracle KEY

the test $\mathscr{U}$ can obtain keys, reflecting that users may use their own keys in detection. The test of course does not have access to the subversion key $\widetilde{K}$. A subversion $\widetilde{\Pi}$ in which this advantage 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. Attacks we will present in Section 4 show that such subversion is possible for a broad class of schemes Π.

We emphasize that the above definition captures the *users*' inability to know which encryption scheme is being used, the real one or the subverted one, even if it knows the private underlying keys. The adversary $\mathscr{U}$ in this setting might be regarded as the good guys—the population of users intent on seeing if they are all being surveilled based on the input/output behavior of the encryption code. We note that even if the detection advantage above is large, it is not clear that users would actually be able to detect subversion: for one thing, they probably wouldn't know what to look for. Thus detection advantage is only interesting when, for a scheme, it is demonstrably small. In that case big-brother has effectively forced detection to work by way of reverse-engineering the subverted code, not by looking at its black-box behavior.

Decryptability of a subversion $\widetilde{\Pi}$ relative to Π is not directly implied by undetectability, as the former is a correctness condition that must hold for all inputs, whereas the latter still allows for algorithms $\mathcal{E}$ and $\widetilde{\mathcal{E}}$ that, for example, produce different outputs only on a single message M that depends on $\widetilde{K}$, with the output of $\widetilde{\mathcal{E}}$ on M failing to decrypt under $\mathcal{D}$. Nor does decryptability imply undetectability, with the construction of a separating example being easy.

Surveillance advantage. Now we want to define what it means for a scheme Π to resist, meaning be secure against, ASAs. The first thought is to ask that big brother, even given its subversion key $\widetilde{K}$, cannot recover the plaintexts underlying subverted ciphertexts. We ask for something stronger, namely that big brother, even given $\widetilde{K}$, cannot tell whether ciphertexts are being produced by the real encryption algorithm $\mathcal{E}$ or by the subverted algorithm $\widetilde{\mathcal{E}}$. Formally let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ be a subversion of it. Let $\mathscr{B}$ be an adversary representing big brother. Let

$$\mathbf{Adv}^{\mathrm{srv}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) \;=\; 2\Pr[\mathrm{SURV}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \Rightarrow \mathsf{true}] - 1$$

where game SURV is shown on the right of Fig. 1. In the game, adversary $\mathscr{B}$ is given the subversion key $\widetilde{K}$, but is not given user keys $K_1, K_2, \ldots$. (We remark that the SURV and DETECT games are very similar, effectively duals of each other, the Enc oracle in particular being the same. The difference is that in the former the adversary gets $\widetilde{K}$ but not $K_1, K_2, \ldots$ while in the latter it is the other way around.) For Π to be secure against surveillance requires that this advantage is small for all subversions $\widetilde{\Pi}$ of Π and all $\mathscr{B}$. This is the desired notion for *positive* results, and we will present schemes secure in this sense in Section 5. (We will assume minimal detection security in the form of the decryptability condition. Without some resistance to detection, surveillance security is not possible.) In offering a scheme secure in this sense we are asserting that big-brother can't come close to achieving surveillance *en masse*.

We have formulated surveillance security with multiple users, but a hybrid argument shows that the advantage relative to the one-user game can grow by at most a factor of the number of users. We will use this result to simplify proofs, which will restrict attention to the game with a single user. We remark that a similar claim is not true for detection security.

4 Mounting ASAs

This section shows that typical randomized, stateless encryption schemes are subvertible. We first describe an attack on modes of operation that surface their IV. Then we describe what we call a universal attack, so named because it applies regardless of the specifics of the scheme being attacked. In Appendix A we explain to what extent such attacks are applicable to the most important secure communications protocols for the Internet, namely SSL/TLS, IPsec, and SSH.

4.1 IV-replacement attacks

Following Young and Yung [44], Goh, Boneh, Pinkas and Golle [16] consider the problem of adding a hidden key recovery to protocols. They suggest that when the server needs a random nonce, it can use in its place an encryption of the session key computed under the escrow key. We expand on this idea, letting the escrow key be the subversion key. We show how to subvert stateless encryption schemes that put a random nonce into the ciphertext.

We consider randomized, stateless schemes $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, writing $C \leftarrow \mathcal{E}(K, M, A; IV)$, where we now surface the randomness input IV (for initial vector, IV) to the encryption algorithm and suppress the state input. Such a scheme is said to surface its IV if there is an efficient algorithm $\mathcal{X}$ such that $\mathcal{X}(\mathcal{E}(K, M, A; IV)) = IV$ for all K, M, A, IV. The condition says that $\mathcal{X}$ can recover the IV from the ciphertext. A simple example of a scheme that surfaces its IV is CBC\$, namely CBC mode with random IV. Another example is CTR\$, counter mode with random starting point.

The first requirement of a subversion attack is undetectability, but other attributes are relevant too. We will describe two attacks.

Stateful attack. This is the simplest attack, in which the IV is simply replaced by an encipherment, under the subversion key $\widetilde{K}$, of the encryption key K. For simplicity of presentation, we assume that the IV length and key length are the same. (The attack extends easily to accommodate cases where the key length is greater than the IV length, simply by leaking K in pieces across multiple ciphertexts, and using state in $\widetilde{\mathcal{E}}$ to keep track of which piece of K is to be leaked in each ciphertext; the opposite case can also be handled by appropriate padding of K prior to its encipherment.) In order to prevent repeated IVs being seen across ciphertexts, we must limit the IV substitution to one ciphertext. This necessitates the use of a stateful subversion scheme. To avoid this repetition, one might consider replacing the IV by the encryption of K under a randomized symmetric encryption scheme that is IND\$-CPA secure, but, since this encryption will usually be longer than the IV and thus cannot replace the IV in a single ciphertext, we would need to adopt a stateful approach to implement it too.

In more detail, let the bit length of the IV and key be n and assume we have a blockcipher $E : \widetilde{\mathcal{K}} \times \{0,1\}^n \to \{0,1\}^n$ with block length n. The subversion of Π is the triple $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ where:

$\underline{\widetilde{\mathcal{E}}(\widetilde{K}, K, M, A, \sigma, i)}$	$\underline{\widetilde{\mathcal{D}}(\widetilde{K}, \boldsymbol{C}, \boldsymbol{A}, i)}$
If $\sigma = 0$ then $IV \leftarrow E(\widetilde{K}, K)$	$IV \leftarrow \mathcal{X}(\boldsymbol{C}[1])$
Else $IV \twoheadleftarrow \{0,1\}^n$	$K \leftarrow E^{-1}(\widetilde{K}, IV)$
$C \leftarrow \mathcal{E}(K, M, A, IV)$	$\boldsymbol{M}[1] \leftarrow \mathcal{D}(K, \boldsymbol{C}[1], \boldsymbol{A}[1])$
$\sigma \leftarrow \sigma + 1$; Return C	Return $\boldsymbol{M}$

The state σ maintained by $\widetilde{\mathcal{E}}$ is an integer initialized at 0. When the state has this initial value, $\widetilde{\mathcal{E}}$ sets the IV to an encryption of the key K, and otherwise performs no subversion, picking the IV at random. Now assume user i has requested an encryption of a message $\boldsymbol{M}[1]$ under associated data $\boldsymbol{A}[1]$ with $\sigma = 0$, resulting in ciphertext $\boldsymbol{C}[1] = \widetilde{\mathcal{E}}(\widetilde{K}, K, \boldsymbol{M}[1], \boldsymbol{A}[1], 0, i)$. The subverter's decryption algorithm gets input $\widetilde{K}$ together with i and the length-one vectors $\boldsymbol{C}, \boldsymbol{A}$, and recovers the key K as shown. Once obtained, the key can be used to decrypt not only the current but any future ciphertexts.

This subversion $\widetilde{\Pi}$ meets the decryptability condition. Furthermore, as long as E is a PRP/PRF, the subverted IV is indistinguishable from a random one, even to an observer that knows K (the observer does not know $\widetilde{K}$), making the subversion undetectable. Formally:

Theorem 1. *Let $\Pi = (\{0,1\}^n, \mathcal{E}, \mathcal{D})$ be a randomized, stateless symmetric encryption scheme that surfaces an IV of length n. Let $E\colon \widetilde{\mathcal{K}} \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let the subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ of Π be defined as above. Let $\mathscr{U}$ be a test that makes q queries to its* Key *oracle. Then we can construct*

an adversary $\mathscr{A}$ such that $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) \le q^2/2^n + \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathscr{A})$. *Adversary $\mathscr{A}$ makes q oracle queries and its running time is that of $\mathscr{U}$.*

The $q^2/2^n$ term corresponds to the chance that two users have the same key, in which case their subverted IVs will be the same while the real ones would be random and independent.

Suppose, however, that a user system, and hence the state of $\widetilde{\mathcal{E}}$, is reset. Then the subverted IV will be recreated and the observer detects a repeated IV, something not likely to happen in the absence of the subversion (though plausibly explainable as a randomness failure). This reduces the effectiveness of this simple attack. One solution to this problem is to adopt the above-mentioned idea of replacing the IV by the encryption of K under a randomized symmetric encryption scheme. This would result in a subversion $(\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ that is both randomized *and* stateful. This subversion would have the practical advantage of being able to continuously leak the key K, rather than relying on big brother to intercept ciphertext $\boldsymbol{C}[1]$. In our next attack, we present a subversion that preserves this property and only requires randomisation.

Stateless attack. We present an attack where $\widetilde{\mathcal{E}}$ is stateless. In this attack the subversion is undetectable even under resets of the encryptor system, making the attack harder to detect in practice. Let k be the key length of Π and let $v = \lceil \log_2(k) \rceil$. (For example if $k = 128$ as for AES then $v = 7$.) Let $E\colon \widetilde{\mathcal{K}} \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher where n is the length of the IV of Π as before. The subversion of Π is the triple $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ where:

$\underline{\widetilde{\mathcal{E}}(\widetilde{K}, K, M, A, i)}$	$\underline{\widetilde{\mathcal{D}}(\widetilde{K}, \boldsymbol{C}, \boldsymbol{A}, i)}$
$\ell \twoheadleftarrow [1..k]$	For $j = 1, \ldots, \lvert\boldsymbol{C}\rvert$ do
$R \twoheadleftarrow \{0,1\}^{n-v-1}$	$\quad b \Vert \ell \Vert R \leftarrow E^{-1}(\widetilde{K}, \mathcal{X}(\boldsymbol{C}[j]))$; $K'[\ell] \leftarrow b$
$IV \leftarrow E(\widetilde{K}, K[\ell] \Vert \ell \Vert R)$	For $j = 1, \ldots, \lvert\boldsymbol{C}\rvert$ do
$C \leftarrow \mathcal{E}(K, M, A, IV)$	$\quad \boldsymbol{M}[j] \leftarrow \mathcal{D}(K', \boldsymbol{C}[j], \boldsymbol{A}[j])$
Return C	Return $\boldsymbol{M}$

In computing $E(\widetilde{K}, K[\ell] \Vert \ell \Vert R)$ the integer ℓ is encoded as a v-bit string. After around $k \ln(k)$ encryptions, we expect that every $\ell \in [1..k]$ has been chosen at least once, so that if a vector of this many ciphertexts is passed to $\widetilde{\mathcal{D}}$, the latter will succeed. Undetectability again follows if E is a PRP/PRF, exploiting the fact that the observer does not know $\widetilde{K}$:

Theorem 2. *Let $\Pi = (\{0,1\}^k, \mathcal{E}, \mathcal{D})$ be a randomized, stateless symmetric encryption scheme that surfaces an IV of length n. Let $E : \widetilde{\mathcal{K}} \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $v = \lceil \log_2(k) \rceil$. Let the subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ of Π be defined as above. Let $\mathscr{U}$ be a test that makes q queries to its* Enc *oracle. Then we can construct an adversary $\mathscr{A}$ such that* $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) \le q^2/2^{n-v-1} + \mathbf{Adv}^{\mathrm{prf}}_{E}(\mathscr{A})$. *It makes q oracle queries and its running time is that of $\mathscr{U}$.*

This subversion achieves an even stronger form of undetectability than Theorem 2 captures. Since the subversion is stateless, reset of the system does not lead to detection. (It is assumed that the subvertor has access to fresh coins at every invocation. If a reset results in re-use of coins, our claim would no longer be true.) The subversion obviously extends to one leaking more than bit of K per ciphertext, at the cost of a weaker bound on detection advantage.

4.2 The biased-ciphertext attack

The above IV-replacement attacks apply to several common modes in their "textbook" form and to some of their deployments in Internet protocols, but there are many encryption schemes to which they do not apply. These include schemes that do not surface the IV, for example encrypted-IV schemes like CBC2 [35], IACBC [23] and XCBC$ [15].

In this section we present a more general attack that we call the biased ciphertext attack. This attack is "universal" in that it applies to any randomized and stateless encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ that uses a minimal amount of randomness, say 7 bits. Undetectability holds in a strong form, namely even under reset of the state of the subverter.

Suppose the user asks its system to use this scheme to encrypt a message M with key K and associated data A, which means that the system is expected to pick coins δ at random from the space D of coins for $\mathcal{E}$ and return ciphertext $C \leftarrow \mathcal{E}(K, M, A; \delta)$ (where we now replace IV by δ to emphasise the fact that δ may not be surfaced). Our subverted encryption algorithm will compute C the same way, except that δ will not be chosen quite at random. Instead, it will be chosen to ensure that $F(\widetilde{K}, C) = K[j]$ is the j-bit of the key, where F is a PRF. The subverter decryption algorithm, on receiving C, will recompute $K[j]$ as $F(\widetilde{K}, C)$. The counter j will be maintained by the subverter algorithms in their state, so that over $|K|$ encryptions, the entire key is leaked. The challenge here is showing that the bias created in the distribution of C is not detectable, *even given the key K.* Exploiting PRF security, we can move to a setting where $F(\widetilde{K}, \cdot)$ is replaced by a random function. Then we use an information-theoretic argument to show that the statistical distance between the real and subverted ciphertexts is small even given K. In terms of our formal definitions, big brother is undetectable.

We highlight the following features of the attack. First, big brother does not pick, or care, what messages or associated data is encrypted – this is no chosen-message attack. Big brother will succeed no matter what the user chooses to encrypt, as long as it encrypts $|K|$ or more messages. Second, the attack does not merely distinguish between real and subverted ciphertexts; rather, it recovers the encryption key. Although presented as a key recovery attack, it is not hard to see that, in terms of our formal definitions, big brother has surveillance advantage close to 1.

Let us say that Π is *coin injective* if the mapping of coins to ciphertext, for each fixed key, message and associated data, is injective. The analysis in our current proof of undetectability requires that Π have this property. The assumption is not particularly restrictive. Schemes that surface their IV are coin injective, not just the ones to which the IV-replacement attack applies, but also ones like OCB with random nonce that, as we indicated, were harder to handle. Schemes that encrypt the IV are also coin injective and thus covered. More generally, our analysis applies when the mapping is not injective but is regular.

Proceeding, suppose $g\colon\ D \to R$ where $D \subseteq \{0,1\}^*$, and $f\colon\ \{0,1\}^* \to \{0,1\}$. For $b \in \{0,1\}$ we let $S^{f,g}(b, D) = \{\delta \in D\colon\ f(g(\delta)) = b\}$. Here think of g as taking coins δ and returning an encryption under them, the key, message, and associated data being fixed as part of g. Let $F\colon\ \widetilde{\mathcal{K}} \times \{0,1\}^* \to \{0,1\}$ be a PRF that returns a bit. The subversion of Π is the triple $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ where:

$\underline{\widetilde{\mathcal{E}}(\widetilde{K}, K, M, A, \sigma, i)}$	$\underline{\widetilde{\mathcal{D}}(\widetilde{K}, \boldsymbol{C}, \boldsymbol{A}, i)}$
$j \leftarrow \sigma \bmod \lvert K\rvert$; $j \leftarrow j + 1$	For $j = 1, \ldots, \lvert\boldsymbol{C}\rvert$ do
$g(\cdot) \leftarrow \mathcal{E}(K, M, A; \cdot) \Vert \sigma \Vert i$	$\quad K'[j] \leftarrow F(\widetilde{K}, \boldsymbol{C}[j] \Vert j-1 \Vert i)$
$\delta \twoheadleftarrow S^{F(\widetilde{K},\cdot),g(\cdot)}(K[j], D)$	For $j = 1, \ldots, \lvert\boldsymbol{C}\rvert$ do
$C \leftarrow \mathcal{E}(K, M, A; \delta)$	$\quad \boldsymbol{M}[j] \leftarrow \mathcal{D}(K', \boldsymbol{C}[j], \boldsymbol{A}[j])$
$\sigma \leftarrow \sigma + 1$; Return C	Return $\boldsymbol{M}$

The state σ maintained by $\widetilde{\mathcal{E}}$ is an integer, initially zero. Encryption lets g be the function that has K, M, A, j, σ, i hardwired and on input coins δ in the space D of coins of $\mathcal{E}$, returns $\mathcal{E}(K, M, A; \delta) \Vert \sigma \Vert i$, the last two components ensuring no collisions in output values of the function across different users and states. Picking δ at random from the indicated set means that the ciphertext $C = \mathcal{E}(K, M, A; \delta)$ will satisfy $F(\widetilde{K}, C \Vert j-1 \Vert i) = K[j]$, except with some probability of error when the set is empty.

Let $k = |K|$. Now assume that user i has requested encryptions of messages $\boldsymbol{M}[1], \ldots, \boldsymbol{M}[k]$ under associated data $\boldsymbol{A}[1], \ldots, \boldsymbol{A}[k]$, respectively, to result in ciphertexts $\boldsymbol{C}[1], \ldots, \boldsymbol{C}[k]$, created via $\boldsymbol{C}[j] =$

$\widetilde{\mathcal{E}}(\widetilde{K}, K, \boldsymbol{M}[j], \boldsymbol{A}[j], j-1, i)$ for $j = 1, \ldots, k$. The big-brother decryption algorithm gets input $\widetilde{K}, \boldsymbol{C}, \boldsymbol{A}, i$ and recovers the key K' as shown. It then decrypts under the true decryption algorithm to return the corresponding vector of messages. Except in the case of an error, the event $K \neq K'$ whose probability we will bound below, not only does decryption succeed, but the process does more, recovering the key, and once this is done the key can be stored and further ciphertexts decrypted directly.

The error probability of the key recovery attack is at most $e_1 + \cdots + e_k$ where $e_j = \Pr[K'[j] \neq K[j]] = \Pr[S^{F(\widetilde{K},\cdot),g(\cdot)}(K[j], D) = \emptyset]$. Assuming F is a good PRF, our estimate can be made with a random function f in its place. Due to the inclusion of $\sigma \| i$ in the argument to f, the applications of f are independent. Assuming g is injective, each time, the set has chance 2^{-d} to be empty where $d = |D|$, so the error probability is at most $k2^{-d}$. This is small as long as the scheme uses a minimal amount of randomness, for example 7 bits, resulting in $d = 2^7 = 128$. (A randomized mode will typically use 96–128 bits of randomness, in which case the error probability is entirely negligible.) A similar analysis can be carried out for the formal surveillance attack.

We claim that the subversion is undetectable. Our analysis first uses the PRF security of F to replace $F(\widetilde{K}, \cdot)$ with a random function f. The key claim is then the following information theoretic lemma.

Lemma 1. *Suppose $g\colon D \to R$. Let $b \in \{0,1\}$ and $\overline{\delta} \in D$. Let $d = |D|$. Let $p = \Pr[\delta = \overline{\delta}]$ where we first draw $f\colon g(D) \to \{0,1\}$ at random and then draw δ at random from $S^{f,g}(b, D) = \{\delta \in D\colon f(g(\delta)) = b\}$.*
(1) *If g is injective then $p = (1 - 2^{-d})/d$.*
(2) *More generally, if g is k-regular, then $p = (1 - 2^{-d/k})/d$.*

Proof (Lemma 1). Let $\overline{\rho} = g(\overline{\delta})$. Then

$$p \;=\; \sum_{T \ni \overline{\rho}} \Pr[f^{-1}(b) = T] \cdot \frac{1}{|g^{-1}(T)|} \;. \tag{1}$$

Here the sum is over all subsets T of $g(D)$ that contain $\overline{\rho}$. The first term in the sum is the probability that T is the set of points that map to b under f. Once T is determined, the second term is the probability that δ chosen at random from $g^{-1}(T)$ equals $\overline{\delta}$. Now, g being injective coupled with T being a subset of $g(D)$ means that $|g^{-1}(T)| = |T|$. Thus we have $p = q$ where

$$q \;=\; \sum_{T \ni \overline{\rho}} \Pr[f^{-1}(b) = T] \cdot \frac{1}{|T|} \;.$$

We now break up the sum according to the number i of points in $T \setminus \{\overline{\rho}\}$. Each particular set T has probability 2^{-r} of equalling $f^{-1}(b)$ where $r = |g(D)|$, so

$$q \;=\; \sum_{i=0}^{r-1} \binom{r-1}{i} 2^{-r} \cdot \frac{1}{i+1} \;.$$

Now we can use some combinatorics to simplify:

$$q \;=\; 2^{-r} \sum_{i=0}^{r-1} \binom{r}{i+1} \frac{1}{r} \;=\; \frac{2^{-r}}{r} \sum_{i=1}^{r} \binom{r}{i} \;=\; \frac{2^{-r}}{r} \cdot (2^r - 1) \;=\; \frac{1 - 2^{-r}}{r} \;. \tag{2}$$

But since g is injective, $r = d$. This proves part (1). For (2), starting from Eq. (1), the k-regularity together with the fact that $T \subseteq g(D)$ means that $|g^{-1}(T)| = k \cdot |T|$ and thus $p = q/k$. On the other hand, $r = d/k$. Combining this with Eq. (2) we have

$$p \;=\; \frac{q}{k} = \frac{1 - 2^{-r}}{kr} = \frac{1 - 2^{-d/k}}{k(d/k)} = \frac{1 - 2^{-d/k}}{d}$$

as claimed.

We use this lemma to estimate the undetectability of the subversion:

Theorem 3. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a randomized, stateless, coin-injective symmetric encryption scheme with randomness-length r, and let $d = 2^r$. Let $F\colon \widetilde{\mathcal{K}} \times \{0,1\}^* \to \{0,1\}$ be a PRF. Let the subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ of Π be defined as above. Let $\mathscr{U}$ be a test that makes q queries to its* Enc *oracle. Then we can construct an adversary $\mathscr{A}$ such that $\mathbf{Adv}^{\mathrm{det}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) \le q/2^d + \mathbf{Adv}^{\mathrm{prf}}_{F}(\mathscr{A})$. Adversary $\mathscr{A}$ makes q oracle queries and its running time is that of $\mathscr{U}$.*

So again as long as the scheme uses a non-trivial amount of randomness, for example $r \ge 7$ bits resulting in $d \ge 128$, Theorem 3 implies that the subversion is undetectable. The proof makes crucial use of Lemma 1, which, letting $D = \{0,1\}^r$ be the space of coins of $\mathcal{E}$, implies that the statistical distance between the real and subverted ciphertexts is

$$\sum_{\delta \in D} \left| \frac{1}{d} - \frac{1 - 2^{-d}}{d} \right| \;=\; d \cdot \frac{2^{-d}}{d} = 2^{-d}\,.$$

A reset of the state will lead to increased detection ability for an observer, but if Π draws its coins from a reasonably large space, this increase does not appear to be enough to lead to actual detection. However the attack continues to be randomized, so if a system reset results in re-use of entropy, detection becomes possible.

5 Defeating ASAs

We turn to finding schemes that resist ASAs. Given the results of Section 4, such schemes must be deterministic and stateful. But not any such scheme works. The challenge here is that security properties of a scheme, such as privacy and authenticity, are of no evident use in showing resistance to ASAs, for these properties hold relative to adversaries that do not know the key K, while in the surveillance game, the subverted encryption algorithm has the key K. Thus surveillance security will rely on combinatorial properties of the scheme. We pinpoint one such property, defining what it means for a symmetric encryption scheme to have unique ciphertexts. We then show that any such scheme is surveillance-resistant. We then present some designs of unique-ciphertext, and thus surveillance-secure, schemes.

Unique ciphertexts. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For any possible state τ of $\mathcal{D}$ with respect to key K, any message $M \in \{0,1\}^*$ and any associated data $A \in \{0,1\}^*$, let $\mathcal{C}_\Pi(K, M, A, \tau)$ be the set of all ciphertexts C such that $\mathcal{D}(K, C, A, \tau)$ accepts with message M, meaning its output is (M, τ') for some τ'. We say that Π has unique ciphertexts if the set $\mathcal{C}_\Pi(K, M, A, \tau)$ has size at most one for all K, M, A, τ. This means that, for any given key, message, associated data and state, there exists at most one ciphertext that the decryptor will decrypt to the message in question.

Due to the correctness condition, any unique-ciphertext scheme is deterministic. The converse is not true, meaning Π being deterministic does not necessarily mean it has unique ciphertexts. If Π is deterministic there is only one ciphertext an honest encryptor will produce given a particular key, message, associated data and state, but determinism does not ensure that there is not some other ciphertext that the decryptor will decrypt to the same message. As an analogy, the difference is the same as between deterministic and unique signature schemes [18, 25]. In the former there is only one signature an honest signer would produce given a signing key and message, but the verifier may accept many signatures for a given verification key and message. In the latter, there exists at most one signature that the verifier accepts for any verification key and message.

Surveillance-security. The following says that a unique-ciphertext scheme cannot be subverted without violating the decryptability condition.

Theorem 4. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a unique ciphertext symmetric encryption scheme. Let $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ be a subversion of Π that obeys the decryptability condition relative to Π. Let $\mathscr{B}$ be an adversary. Then $\mathbf{Adv}^{\mathrm{srv}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) = 0$.*

Proof (Theorem 4). As we remarked in Section 3, a hybrid argument shows that, in game SURV, one may restrict attention to a single user. We thus assume $\mathscr{B}$ first queries $\textsc{Key}(0)$, this being its only query to this oracle, and all its subsequent queries are of the form $\textsc{Enc}(\cdot,\cdot,0)$. It thus suffices to show that $\Pr[\mathrm{G}^{\mathscr{A}} \Rightarrow \mathsf{true}] = 0$ for all $\mathscr{A}$, where we define game G as follows:

Game $\mathrm{G}^{\mathscr{A}}$	$\underline{\textsc{Enc}(M, A)}$
$K \twoheadleftarrow \mathcal{K}$; $\widetilde{K} \twoheadleftarrow \widetilde{\mathcal{K}}$; $b \twoheadleftarrow \{0,1\}$; $j \leftarrow 0$	$j \leftarrow j+1$; $A_j \leftarrow A$; $M_j \leftarrow M$
$\sigma_{0,0}, \sigma_{1,0} \leftarrow \varepsilon$; $\tau_{0,0}, \tau_{1,0} \leftarrow \varepsilon$	$(C_{1,j}, \sigma_{1,j}) \leftarrow \mathcal{E}(K, M_j, A_j, \sigma_{1,j-1})$
$b' \twoheadleftarrow \mathscr{A}^{\textsc{Enc}}$	$(C_{0,j}, \sigma_{0,j}) \leftarrow \widetilde{\mathcal{E}}(\widetilde{K}, K, M_j, A_j, \sigma_{0,j-1}, 0)$
Return $(b = b')$	$(M_{1,j}, \tau_{1,j}) \leftarrow \mathcal{D}(K, C_{1,j}, A_j, \tau_{1,j-1})$
	$(M_{0,j}, \tau_{0,j}) \leftarrow \mathcal{D}(K, C_{0,j}, A_j, \tau_{0,j-1})$
	Return $C_{b,j}$

Let $M_{0,0} = M_{1,0} = C_{0,0} = C_{1,0} = \varepsilon$. We claim that $(M_{1,j}, C_{1,j}, \tau_{1,j}) = (M_{0,j}, C_{0,j}, \tau_{0,j})$ for all $j = 0, \ldots, q$ where q is the number of $\textsc{Enc}$ queries of $\mathscr{A}$. If this is true then the ciphertexts returned by $\textsc{Enc}$ do not depend on b, which means that $\Pr[\mathrm{G}^{\mathscr{A}} \Rightarrow \mathsf{true}] = 0$. So it remains to establish the claim, which we do by induction on j. The base case is $j = 0$, where the claim is true because all quantities involved equal ε. Now suppose $1 \le j \le q$ and assume $(M_{1,j-1}, C_{1,j-1}, \tau_{1,j-1}) = (M_{0,j-1}, C_{0,j-1}, \tau_{0,j-1})$. We will show that $(M_{1,j}, C_{1,j}, \tau_{1,j}) = (M_{0,j}, C_{0,j}, \tau_{0,j})$. The induction hypothesis implies that $\tau_{0,j-1} = \tau_{1,j-1}$ and we denote this common value by τ_{j-1}. The assumption that $\widetilde{\Pi}$ satisfies the decryptability condition implies that $M_{0,j} = M_j$. The correctness of Π implies that $M_{1,j} = M_j$. Thus $M_{0,j} = M_{1,j} = M_j \neq \perp$. This means that $C_{0,j}$ and $C_{1,j}$ both belong to the set $\mathcal{C}(K, M_j, A_j, \tau_{j-1})$. But the unique ciphertext assumption on Π says that this set has size at most one so it must be that $C_{1,j} = C_{0,j}$. The determinism of $\mathcal{D}$ now implies that $\tau_{0,j} = \tau_{1,j}$, concluding the induction.

A UNIQUE-CIPHERTEXT SCHEME. We give an example of a symmetric encryption scheme that has unique ciphertexts and hence, by Theorem 4, is not subvertible. Our scheme is based on the encode-then-encipher paradigm of [7] which we extend to allow associated data. Let $P\colon \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$ be a family of permutations. This means that P_K is injective and length-preserving ($|P_K(x)| = |x|$ for all $x \in \{0,1\}^*$) for every $K \in \{0,1\}^k$. By P^{-1} we denote the inverse of P, satisfying $P_K^{-1}(P_K(x)) = x$ for all $x \in \{0,1\}^*$. We also let $F\colon \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^t$ be a family of functions. (It will be used as a MAC.) The state σ in our scheme will be a counter, and we denote by $\langle\sigma\rangle$ its representation as a ℓ-bit string. Our symmetric encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ has key space $\mathcal{K} = \{0,1\}^{2k}$ and encryption and decryption algorithms defined as follows:

$\underline{\mathcal{E}(K, M, A, \sigma)}$	$\underline{\mathcal{D}(K, C, A, \tau)}$
If $(\sigma = 2^\ell)$ then return $(\perp, \sigma)$	If $(\tau = 2^\ell)$ then return $(\perp, \tau)$
$K_1 \| K_2 \leftarrow K$	$K_1 \| K_2 \leftarrow K$; $(W, T) \leftarrow C$; $x \leftarrow P^{-1}(K_1, W)$
$W \leftarrow P(K_1, \langle\sigma\rangle \| M)$	If $(\|x\| < \ell)$ then return $(\perp, \tau)$
$T \leftarrow F(K_2, W \| A)$	$\langle\sigma\rangle \| M \leftarrow x$
$C \leftarrow (W, T)$	If $(T \neq F(K_2, W \| A))$ then return $(\perp, \tau)$
$\sigma \leftarrow \sigma + 1$	If $(\sigma \neq \tau)$ then return $(\perp, \tau)$
Return (C, σ)	$\tau \leftarrow \tau + 1$; Return (M, τ)

In the 4th line of the code of $\mathcal{D}$, we are interpreting the first ℓ bits of x as the binary encoding of an integer denoted σ, and letting M be the rest of the bits of x. If P is a PRP and F is a PRF then Π is a secure authenticated encryption scheme. This is a standard claim that can be proved following [7]. Of interest in our context is instead the following, which says that Π has unique ciphertexts. This makes no security assumptions on P or F.

Theorem 5. *Let $P\colon \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^*$ be a family of permutations and $F\colon \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^t$ a family of functions. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the symmetric encryption scheme associated to them as above. Then Π satisfies the correctness condition and has unique ciphertexts.*

Proof. It is clear that Π satisfies the correctness condition. We now show it has unique ciphertexts. Let $K = K_1 \| K_2 \in \{0,1\}^{2k}$ be a key. Let $M, A \in \{0,1\}^*$ and let $\tau \in [0..2^\ell - 1]$. Let $W^* = P(K_1, \langle \tau \rangle \| M)$ and $T^* = F(K_2, W \| A)$. We claim that $\mathcal{C}_\Pi(K, M, A, \tau) = \{(W^*, T^*)\}$, meaning this ciphertext is the only member of the set. This shows the set has size at most one. To prove the claim, assume $C = (W, T)$ is an arbitrary member of $\mathcal{C}_\Pi(K, M, A, \tau)$. By definition of the set, it must be that $\mathcal{D}(K, C, A, \tau)$ returns (M, τ') for some τ'. But then from the code of $\mathcal{D}$, and the fact that P is a permutation and F is deterministic, we must have $W = P(K_1, \langle \tau \rangle \| M)$ and $T = F(K_2, W \| A)$.

Surveillance-resistance from nonce-based schemes. Above we gave a simple scheme to illustrate that surveillance-resistance is possible. However, likely candidates to instantiate the PRP are two pass [19, 20], making the scheme potentially slower than standard, deployed ones. Here we describe a better solution. We show that any nonce-based scheme meeting a natural non-degeneracy condition, called "tidiness" in [28], can be turned into a stateful symmetric encryption scheme (by using the nonce as a counter) that has unique ciphertexts. Most existing and practical nonce-based schemes meet our condition, so this results in a number of surveillance-secure schemes that may be easily deployed. Proceeding to the details, we recall that a nonce-based encryption scheme [34, 35] is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Here $\mathcal{K}$ is a finite, non-empty set. $\mathcal{E}$ is a deterministic algorithm that given key K, message M, associated data A and nonce N returns ciphertext $C \leftarrow \mathcal{E}(K, M, A, N)$. $\mathcal{D}$ is a deterministic algorithm that given K, C, A, N returns $\mathcal{D}(K, C, A, N) \in \{0,1\}^* \cup \{\perp\}$. Correctness requires that $\mathcal{D}(K, \mathcal{E}(K, M, A, N), A, N) = M$ for all $K \in \mathcal{K}$ and all M, A, N. Privacy for such schemes holds even when the adversary chooses the nonce as long as the noce does not repeat and is formalized in [34]. We associate to nonce-based scheme Π a stateful symmetric encryption scheme $\Pi^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ as follows. Let $\mathcal{K}^* = \mathcal{K}$ and define

$$\begin{array}{l|l} \underline{\mathcal{E}^*(K, M, A, \sigma)} & \underline{\mathcal{D}^*(K, C, A, \tau)} \\ \text{Return } (\mathcal{E}(K, M, A, \sigma), \sigma + 1) & \text{Return } (\mathcal{D}(K, C, A, \tau), \tau + 1) \end{array}$$

Let us say that nonce-based scheme Π is non-degenerate if for every K, M, A, N there is at most one C such that $\mathcal{D}(K, C, A, N) = M$. Then we trivially have the following.

Theorem 6. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a non-degenerate nonce-based scheme and let $\Pi^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$ be the stateful symmetric encryption scheme obtained from Π as above. Then Π^* satisfies the correctness condition and has unique ciphertexts.*

Non-degenerate nonce-based schemes include OCB [36] and several modes from [34, 35]. Any of these yield a surveillance-secure symmetric encryption scheme via the above.

Acknowledgments

Bellare was supported in part by NSF grants CNS-1228890 and CNS-1116800, Paterson by EPSRC Leadership Fellowship EP/H005455/1, and Rogaway by NSF grants CNS-1228828 and CNS-1314885.

References

1. M. Albrecht, K. Paterson, and G. Watson. Plaintext recovery attacks against SSH. In *IEEE Symposium on Security and Privacy*, pages 16–26. IEEE Computer Society, 2009.
2. N. AlFardan, D. Bernstein, K. Paterson, B. Poettering, and J. C. Schuldt. On the security of RC4 in TLS and WPA. In *USENIX Security Symposium*, 2013.

3. J. Ball, J. Borger, and G. Greenwald. Revealed: how US and UK spy agencies defeat internet security and privacy. The Guardian, `http://www.theguardian.com/world/2013/sep/05/nsa-gchq-encryption-codes-security`, Sep. 2013.
4. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, Oct. 1997.
5. M. Bellare, T. Kohno, and C. Namprempre. Authenticated encryption in SSH: Provably fixing the SSH binary packet protocol. In V. Atluri, editor, *ACM CCS 02*, pages 1–11. ACM Press, Nov. 2002.
6. M. Bellare, T. Kohno, and C. Namprempre. The Secure Shell (SSH) Transport Layer Encryption Modes. RFC 4344 (Proposed Standard), Jan. 2006.
7. M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 317–330. Springer, Dec. 2000.
8. S. Cabuk, C. Brodley, and C. Shields. IP covert channel detection. *ACM Trans. Inf. Syst. Secur.*, 12(4), 2009.
9. S. Checkoway, M. Fredrikson, R. Niederhagen, A. Everspaugh, M. Green, T. Lange, T. Ristenpart, D. J. Bernstein, J. Maskiewicz, and H. Shacham. On the practical exploitability of Dual EC in TLS implementations. In *USENIX Security Symposium*, 2014.
10. T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), Jan. 1999. Obsoleted by RFC 4346, updated by RFCs 3546, 5746, 6176.
11. T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346 (Proposed Standard), Apr. 2006. Obsoleted by RFC 5246, updated by RFCs 4366, 4680, 4681, 5746, 6176.
12. T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), Aug. 2008. Updated by RFCs 5746, 5878, 6176.
13. T. Duong and J. Rizzo. Here come the $\oplus$ Ninjas. Unpublished manuscript, 2011.
14. A. Freier, P. Karlton, and P. Kocher. The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Historic), Aug. 2011.
15. V. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. In M. Matsui, editor, *FSE 2001*, volume 2355 of *LNCS*, pages 92–108. Springer, Apr. 2001.
16. 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.
17. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
18. S. Goldwasser and R. Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In E. F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 228–245. Springer, Aug. 1992.
19. S. Halevi and P. Rogaway. A tweakable enciphering mode. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499. Springer, Aug. 2003.
20. S. Halevi and P. Rogaway. A parallelizable enciphering mode. In T. Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 292–304. Springer, Feb. 2004.
21. R. Housley. Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP). RFC 4309 (Proposed Standard), Dec. 2005.
22. K. Igoe and J. Solinas. AES Galois Counter Mode for the Secure Shell Transport Layer Protocol. RFC 5647 (Informational), Aug. 2009.
23. C. Jutla. Encryption modes with almost free message integrity. *Journal of Cryptology*, 21(4):547–578, Oct. 2008.
24. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, Aug. 1996.
25. A. Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In M. Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 597–612. Springer, Aug. 2002.
26. D. McGrew and D. Bailey. AES-CCM Cipher Suites for Transport Layer Security (TLS). RFC 6655 (Proposed Standard), July 2012.
27. J. Millen. 20 years of covert channel modeling and analysis. In *IEEE Symposium on Security and Privacy*, pages 113–114, 1999.
28. C. Namprempre, P. Rogaway, and T. Shrimpton. Reconsidering generic composition. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 257–274. Springer, 2014.
29. 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.
30. 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.

31. R. Pereira and R. Adams. The ESP CBC-Mode Cipher Algorithms. RFC 2451 (Proposed Standard), Nov. 1998.
32. 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.
33. P. Rogaway. Problems with proposed IP cryptography. Unpublished manuscript, 1995. `http://www.cs.ucdavis.edu/~rogaway/papers/draft-rogaway-ipsec-comments-00.txt`.
34. P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *ACM CCS 02*, pages 98–107. ACM Press, Nov. 2002.
35. P. Rogaway. Nonce-based symmetric encryption. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 348–359. Springer, Feb. 2004.
36. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *ACM CCS 01*, pages 196–205. ACM Press, Nov. 2001.
37. J. Salowey, A. Choudhury, and D. McGrew. AES Galois Counter Mode (GCM) Cipher Suites for TLS. RFC 5288 (Proposed Standard), Aug. 2008.
38. 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.
39. D. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *USENIX Security Symposium*, 2001.
40. K. Thompson. Reflections on trusting trust. *Commun. ACM*, 27(8):761–763, 1984.
41. J. Viega and D. McGrew. The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP). RFC 4106 (Proposed Standard), June 2005.
42. T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. RFC 4253 (Proposed Standard), Jan. 2006. Updated by RFC 6668.
43. 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.
44. 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.
45. A. Young and M. Yung. Kleptography: Using cryptography against cryptography. In *EUROCRYPT*, pages 62–74, 1997.
46. 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.
47. 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.
48. 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.

A Subversion of Internet Symmetric Encryption Protocols

We examine the symmetric encryption components of three major secure Internet protocols, SSL/TLS, IPsec, and SSH, to determine the extent to which they are vulnerable to ASAs.

SSL/TLS. CBC mode in SSL 3.0 [14] and TLS 1.0 [10] uses a chained IV, meaning that the IV for encrypting a message is taken as the last block of the previous ciphertext. This makes the IV predictable, rendering these versions of SSL/TLS vulnerable to an attack first pointed out in [33], which in due course evolved into the BEAST plaintext recovery attack [13]. For the first message on a connection, the IV is derived from the SSL/TLS master secret by applying a key derivation function. This initial IV is not sent on the wire, but instead the receiver also computes it from the SSL/TLS master secret. Thus SSL 3.0 and TLS 1.0 do not surface the CBC mode IV and thus appear invulnerable to the IV replacement attacks of Section 4.1.

However, SSL 3.0's use of random padding and TLS 1.0's support for variable length padding renders these versions vulnerable to the biased ciphertext attack presented in Section 4.2. In more detail, SSL/TLS uses a MAC-pad-encrypt construction; SSL 3.0 uses padding to bring the size p of the combined plaintext and MAC field up to a multiple of the block size b prior to encryption. The padding consists of an

arbitrary sequence of bytes followed by a single length byte, and is at most b bytes in length. This means that, except in a single case (where $p \bmod b = b-1$), there is at least one byte of freedom in the padding. TLS 1.0 again uses padding, but the padding now has a fixed format. However, *variable* length padding is now permitted – anywhere between 1 and 256 bytes of padding can be added, as long as the result of adding it aligns on a block boundary. There are then $d = \log_2(256/b)$ bits of flexibility in choosing the padding. When $b = 16$ (AES, say), this amount to 4 bits per ciphertext. This is slightly less than one would like for the reliable application of the biased ciphertext attack, but that attack is easily extended to add greater resilience to errors.

CBC mode in TLS 1.1 [11] and TLS 1.2 [12] uses an explicit, per packet IV, directly rendering these versions of TLS vulnerable to IV replacement attack.

TLS 1.2 also supports Authenticated Encryption modes, with AES-GCM and AES-CCM for TLS 1.2 being defined in [37] and [26], respectively. The former involves the use of a 12-byte IV (or nonce), of which 8 bytes are carried in each TLS record in the `GenericAEADCipher.nonce_explicit` field. The specification requires this 8-byte value to be unique for each TLS record encrypted under a fixed key, and suggests that the TLS Record Protocol sequence number may be used. However, it does not mandate this use. This flexibility means that an implementation of AES-GCM in TLS 1.2 can be RFC-compliant but still vulnerable to an IV replacement attack. The same situation pertains for AES-CCM in TLS 1.2: the specification uses exactly the same nonce construction as for AES-GCM, and 8 bytes of it are carried explicitly in each TLS record.

The RC4 stream cipher is also an option in all versions of SSL/TLS. Here, the algorithm is stateful, with the RC4 internal state being carried over from one message to the next on a TLS connection. RC4 does not surface an IV, but can no longer be considered secure for use in general purpose applications of TLS in view of recent attacks [2].

IPsec. CBC mode for IPsec is described in [31]. It uses explicit, per packet IVs, and these are recommended to be random. This makes it vulnerable to an IV replacement attack. AES-GCM for IPsec is described in [41]. It involves an 8 byte IV which is transmitted for every packet. According to [41], the only requirement is that the IV be unique, though using a counter is mentioned as the most natural way to achieve this property. This flexibility means that an implementation of AES-GCM in IPsec can be RFC-compliant but still vulnerable to an IV replacement attack. AES-CCM for IPsec is defined in [21]. It again uses an 8-byte IV, which must be unique but which the sender chooses at will. This offers the same flexibility as for AES-GCM in IPsec, and the same attack vector.

SSH. CBC mode for SSH as defined in [42] uses chained IVs (and is therefore does not achieve privacy [33]). This mode for SSH was also broken in [1]. While it does not surface its nonce, the use of random padding by this mode in SSH makes it vulnerable to the biased ciphertext attack. More specifically, each SSH packet includes between 4 and 255 bytes of padding, and this padding should be random [42, Section 6].

CTR mode for SSH is defined in [6, Section 4]. It makes use of an internal counter whose initial value is determined by hashing a key determined in the SSH key exchange protocol. This counter is not transmitted as part of ciphertexts. This means that CTR mode in SSH does not surface its IV, and so appears to be invulnerable to IV replacement attack. However, its support for random padding again makes it vulnerable to the biased ciphertext attack.

AES-GCM for SSH is defined in [22]. It makes use of an 8-byte field called the invocation counter, which is incremented for each packet sent. The specification does not mandate how the initial value of the invocation counter should be set, but we have been informed that, at least in the leading OpenSSH implementation, it is derived from values established securely in the key exchange phase of SSH. It is included as part of the 12-byte IV that is consumed by the AES-GCM algorithm. It is not sent on the wire as part of SSH packets. This would suggest that AES-GCM for SSH is not vulnerable to IV replacement

Game DETECT2$^{\mathscr{U}}_{\Pi,\widetilde{\Pi}}$	**Game** SURV2$^{\mathscr{B}}_{\Pi,\widetilde{\Pi}}$
$b \twoheadleftarrow \{0,1\}$; $(\widetilde{K},\widetilde{L}) \twoheadleftarrow \widetilde{\mathcal{K}}$; $b' \twoheadleftarrow \mathscr{U}^{\text{KEY},\text{ENC}}(\widetilde{K})$	$b \twoheadleftarrow \{0,1\}$; $(\widetilde{K},\widetilde{L}) \twoheadleftarrow \widetilde{\mathcal{K}}$; $b' \twoheadleftarrow \mathscr{B}^{\text{KEY},\text{ENC}}(\widetilde{L})$
Return $(b = b')$	Return $(b = b')$
KEY(i)	KEY(i)
If $(K_i = \bot)$ then $K_i \twoheadleftarrow \mathcal{K}$; $\sigma_i \leftarrow \varepsilon$	If $(K_i = \bot)$ then $K_i \twoheadleftarrow \mathcal{K}$; $\sigma_i \leftarrow \varepsilon$
Return K_i	Return ε
ENC (M,A,i)	ENC (M,A,i)
If $(K_i = \bot)$ then return $\bot$	If $(K_i = \bot)$ then return $\bot$
If $(b = 1)$ then $(C,\sigma_i) \twoheadleftarrow \mathcal{E}(K_i,M,A,\sigma_i)$	If $(b = 1)$ then $(C,\sigma_i) \twoheadleftarrow \mathcal{E}(K_i,M,A,\sigma_i)$
Else $(C,\sigma_i) \twoheadleftarrow \widetilde{\mathcal{E}}(\widetilde{K},K_i,M,A,\sigma_i,i)$	Else $(C,\sigma_i) \twoheadleftarrow \widetilde{\mathcal{E}}(\widetilde{K},K_i,M,A,\sigma_i,i)$
Return C	Return C

Fig. 2. Games used to define the detection and surveillance security of the asymmetric subversion $\widetilde{\Pi} = (\widetilde{\mathcal{K}},\widetilde{\mathcal{E}},\widetilde{\mathcal{D}})$ of encryption scheme $\Pi = (\mathcal{K},\mathcal{E},\mathcal{D})$.

attacks. However, this mode of SSH is still vulnerable to the biased ciphertext attack because of its use of random padding.

B Asymmetric Subversion

In our notion of subversion, compromise of the subversion key $\widetilde{K}$ provides users with the same cryptographic capabilities as big brother; if $\mathscr{B}$ can strip away all encryption by knowing $\widetilde{K}$, so too can someone who reverse engineers it from a deployed cryptosystem. It is possible that adversary $\mathscr{B}$ does not want this. One option is that it use many different keys $\widetilde{K}$ so that any particular compromise has limited consequences. Another possibility, following Young and Yung's descriptions of kleptography [45], is that the subverted encryption use asymmetric rather than symmetric encryption so that compromise of the system reveals only a public encryption key, $\mathscr{B}$ alone holding the corresponding decryption key. The issue is relatively easy to deal with, in terms of definitions, attacks, and provably-sound schemes. Here we sketch the first of these, replacing the master key $\widetilde{K}$ with a pair of master keys $(\widetilde{K},\widetilde{L})$. The full version discusses modification to attacks. Our ASA-resistant schemes are also ASA resistant with respect to the asymmetric ASA notions given below.

MODIFIED SYNTAX. Let $\Pi = (\mathcal{K},\mathcal{E},\mathcal{D})$ be a symmetric encryption scheme. An *asymmetric subversion* of Π is a triple $\widetilde{\Pi} = (\widetilde{\mathcal{K}},\widetilde{\mathcal{E}},\widetilde{\mathcal{D}})$. The syntax is as before except that $\widetilde{\mathcal{K}}$, the *master-key generation algorithm*, is now a probabilistic algorithm that outputs a pair of strings, $(\widetilde{K},\widetilde{L})$, which we call the *deployed master key* and the *private master key*. The subverted encryption algorithm $\widetilde{\mathcal{E}}$ is unchanged; in particular, it still maps $(\widetilde{K},K,M,A,\sigma,i)$ to a pair of strings (C,σ'). But the *plaintext-recovery algorithm* $\widetilde{\mathcal{D}}$ now takes in $\widetilde{L}$ (instead of $\widetilde{K}$) and $\boldsymbol{C},\boldsymbol{A},i$.

MODIFIED SECURITY DEFINITIONS. See Fig. 2 for the games used in defining the asymmetric variants of detection and surveillance advantage. Let $\Pi = (\mathcal{K},\mathcal{E},\mathcal{D})$ be an encryption scheme and let $\widetilde{\Pi} = (\widetilde{\mathcal{K}},\widetilde{\mathcal{E}},\widetilde{\mathcal{D}})$ be an asymmetric subversion of it. Let $\mathscr{U}$ be an algorithm representing a detection test (perhaps hackers, concerned citizens, or a rival intelligence agency). Define

$$\mathbf{Adv}^{\text{det2}}_{\Pi,\widetilde{\Pi}}(\mathscr{U}) \;=\; 2\Pr[\text{DETECT2}^{\mathscr{U}}_{\Pi,\widetilde{\Pi}} \Rightarrow \mathsf{true}] - 1$$

where game DETECT2 is shown on the left of Fig. 2. This measures the ability $\mathscr{U}$ to detect use of an ASA even when the deployed master key is known to party $\mathscr{U}$. As before, entity $\mathscr{U}$ may not make a query $\textsc{Enc}(\cdot,\cdot,i)$ unless it has earlier queried $\textsc{Key}(i)$. A subversion $\widetilde{\Pi}$ in which the advantage above is negligible for all practical $\mathscr{U}$ would be one that evades detection in an increasingly powerful sense: even if some server is known to use a deployed master key $\widetilde{K}$, it remains infeasible to determine if another server runs the honest code or code subverted with the exact same key.

Next we to define what it means for a scheme Π to resist asymmetric surveillance. Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme and let $\widetilde{\Pi} = (\widetilde{\mathcal{K}}, \widetilde{\mathcal{E}}, \widetilde{\mathcal{D}})$ be an asymmetric subversion of it. Let $\mathscr{B}$ be an adversary representing big brother. Let

$$\mathbf{Adv}^{\mathrm{srv2}}_{\Pi,\widetilde{\Pi}}(\mathscr{B}) \;=\; 2\Pr[\mathrm{SURV2}^{\mathscr{B}}_{\Pi,\widetilde{\Pi}} \Rightarrow \mathsf{true}] - 1$$

where game SURV2 is shown on the right of Fig. 2. In the game, adversary $\mathscr{B}$ is given the private subversion key $\widetilde{L}$ (which can be assumed to include the deployed subversion key $\widetilde{K}$), but it is not given user keys $K_1, K_2, \ldots$.