Security of Symmetric Encryption in the Presence of Ciphertext Fragmentation[*]

Alexandra Boldyreva[1†], Jean Paul Degabriele[2‡], Kenneth G. Paterson[2§], and Martijn Stam[3]

[1] Georgia Institute of Technology
[2] Royal Holloway, University of London
[3] University of Bristol

Abstract. In recent years, a number of standardized symmetric encryption schemes have fallen foul of attacks exploiting the fact that in some real world scenarios ciphertexts can be delivered in a fragmented fashion. We initiate the first general and formal study of the security of symmetric encryption against such attacks. We extend the SSH-specific work of Paterson and Watson (Eurocrypt 2010) to develop security models for the fragmented setting. We also develop security models to formalize the additional desirable properties of ciphertext boundary hiding and robustness against Denial-of-Service (DoS) attacks for schemes in this setting. We illustrate the utility of each of our models via efficient constructions for schemes using only standard cryptographic components, including constructions that simultaneously achieve confidentiality, ciphertext boundary hiding and DoS robustness.

1 Introduction

Despite the existence of proofs guaranteeing security, deployed schemes do get compromised sometimes. Consider for example SSH, one of the most widely used secure protocols. Bellare et al. [4] have formally analysed variants of SSH's Binary Packet Protocol (BPP) and showed that these variants are secure. Yet a few years later, Albrecht et al. [1] presented plaintext recovery attacks against these provably secure SSH BPP variants. These attacks exploited the fact that encrypted data can be delivered to the receiver in a fragmented, byte-by-byte manner, and that the attacker can observe the receiver's behaviour at each point (in particular how long it takes to reject certain carefully crafted faulty ciphertexts). On the other hand, formal security definitions, including the one used

[*] This work has been supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II.

[†] This author is supported in part by NSF CAREER award 0545659 and NSF Cyber Trust award 0831184.

[‡] This author is supported by Vodafone Group Services Limited, a Thomas Holloway Research Studentship, and the Strategic Educational Pathways Scholarship Scheme (Malta), part-financed by the European Union European Social Fund.

[§] This author is supported by EPSRC Leadership Fellowship EP/H005455/1.

to prove SSH secure, traditionally treat plaintexts and ciphertexts as *atomic*, meaning that the entire ciphertext is offered for decryption and a plaintext (or error symbol) is instantly returned.

To bridge this gap between theory and practice and to have schemes with security guarantees that hold not only on paper but also in reality, one has to design security definitions which are integrated better with the environments in which the protocols are deployed. Paterson and Watson [14] recently took a first step in this direction by showing that certain SSH BPP variants meet a newly introduced security notion that takes the aforementioned attacks into account. However, their security definition itself is heavily intertwined with the SSH BPP specification and too complex to be extended easily to apply to different schemes. We provide a more detailed critique of this precursor [14] in the full version [6].

Overview of Contributions. In this work we initiate a general study of security of symmetric encryption schemes against fragmentation attacks. Our study goes beyond just message privacy, and also includes length-hiding (or, more precisely, hiding ciphertext boundaries in a ciphertext stream) and the prevention of fragmentation-enabled Denial-of-Service (DoS) attacks against the receiver. These two properties have not been previously studied, partly because the corresponding threats are not present if encryption is treated as being atomic. The framework we develop can be used to provide *meaningful* provable security analyses of practical schemes when deployed in environments that permit ciphertext fragmentation attacks (including but not limited to the ones from [14]).

We complement our new security definitions with efficient cryptographic constructions based on standard primitives meeting the new goals. While it may be relatively easy to achieve each security goal independently, it transpires that it is not straightforward to achieve two or three of the aforementioned goals simultaneously and one of our schemes is the first to do so.

Let us now describe our focus and results in a little more detail.

Data fragmentation. Data sent over networks is often fragmented, meaning that it is broken up into smaller pieces, or packets. If the data is encrypted, the receiver first has to determine what constitutes a complete ciphertext in order to decrypt it and obtain the underlying message. Reconstruction of the original ciphertext by the receiver can be accomplished by various methods. For example, SSH uses a length field that tells the receiver how many bytes are needed before the complete ciphertext has arrived; this length field is encrypted, ostensibly to increase the security of the protocol against traffic analysis.

During transmission, the packets can be accidentally delayed or delivered out of order. But there also may be malicious tampering of legitimate fragmentation, such as breaking the encrypted data into adversarially selected packets, or maliciously delaying their delivery. We already mentioned an attack of this kind on SSH [1]. Another example is an attack by Degabriele and Paterson [8] on the IPsec protocol. While fragmentation is used adversarially in both attacks, there are some notable differences. In particular, in SSH for honest users, the ciphertext can be effectively regarded as a bitstring (the result of say CBC-and-MAC) and it is only the adversary who starts fragmenting this string. In IPsec, further protocol layers already forcibly introduce fragmentation; on the one hand this ties the adversary, but on the other hand the interaction of this protocol layer with the cryptographic layer can offer new avenues of attack (as exploited by the attack on IPsec [8]). Our treatment addresses both scenarios.

Syntax for encryption supporting fragmentation. We start our analysis with defining encryption in the presence of fragmentation. A *symmetric encryption scheme supporting fragmentation* is defined similarly to a regular atomic (fragmentation devoid) encryption scheme, except the decryption algorithm is always stateful, mainly to model decryption algorithms that may, for example, combine data coming from multiple ciphertext fragments before outputting any plaintext. In addition to state, decryption takes input fragments, one-by-one. Depending on the scheme, the minimal fragment length can be one bit, a byte or a block (of some fixed length). The correctness requirement is defined more intricately than that for atomic encryption. It requires that regardless of how one fragments the ciphertexts, the original messages are returned, with correct message boundaries indicated.

In this paper we focus on two subclasses of symmetric encryption schemes for fragmented ciphertexts, for which we give separate security definitions. One subclass, to which we will simply refer to as *stateful*, covers most practical encryption schemes, whose encryption and decryption algorithms are both stateful. For example, encryption and decryption can both use a counter that increases depending on the number of messages or ciphertexts processed.

The other subclass we consider is an extension of standard (atomic) encryption schemes that makes handling fragmented ciphertexts possible. Namely, the decryption algorithm is now stateful, but the state just models the buffer the receiver keeps to store the ciphertext fragments before a complete ciphertext is received (thus allowing the decryption oracle to perform operations on the entire ciphertext, even if it arrives in fragments). We call such schemes *stateless beyond buffering (sbb).* Because of space constraints, we focus here on the stateful case, with details for the sbb case appearing in the full version [6].

Message privacy in the presence of fragmentation. We observe that fragmentation becomes relevant for security only in the case of chosen-ciphertext attacks (CCA). We extend the existing CCA security notions for regular atomic (IND-CCA) and atomic and stateful (IND-sfCCA [4]) schemes to the case of ciphertext fragmentation (denoted CFA).

Recall that for IND-sfCCA there is no restriction on the decryption queries, but if the adversary forwards the challenge ciphertexts returned by the left-or-right encryption oracle to the decryption oracle in order, this is considered in-sync, and the decryption output is suppressed. Otherwise, it is declared out-of-sync, and the decryptions are returned to the adversary. This allows the adversary to advance the state of both encryption and decryption algorithms to potentially favourable values. When dealing with fragments, the challenge is to decide when to enter the out-of-sync state. We found the need to declare part of the fragment in-sync, and part of it out-of-sync and resolve the ambiguity with regards to the exact boundary to use. We provide our IND-sfCFA definition and more discussion in Section 3.2.

Ciphertext boundary hiding. It is conventional wisdom in cryptographic security definitions that an encryption scheme is allowed to leak the length of the ciphertext; it is often regarded as inevitable. However, real schemes try to achieve another goal as well: they try to hide the lengths of encrypted messages, with a view to frustrating traffic analysis based on these lengths. This is generally achieved in practice by two distinct mechanisms.

Firstly, an encryption scheme for which the ciphertext length does *not* deterministically depend on the message length may be used (e.g. by using variable-length padding). The SSH Binary Packet Protocol and the TLS Record Protocol both adopt this approach. This mechanism has recently received attention from differing perspectives [16, 18]. Secondly, an encryption scheme may be designed in such a way that it is hard to distinguish where the boundaries between ciphertexts lie in a *stream* of ciphertexts. TLS, with its explicit length field in the header of each TLS Record Protocol message, does not achieve this. But SSH's Binary Packet Protocol (BPP) does attempt to achieve boundary hiding. This necessitated the introduction of an encrypted length field in SSH, which is used by the receiver to determine how many bytes are required before a complete ciphertext has arrived and a MAC on the plaintext can be checked. However, this design decision, coupled with the use of CBC mode encryption, is precisely what enabled recent fragmentation attacks against SSH [1]. Thus having boundary hiding as a security goal can act in opposition to achieving other, more standard security goals.

In Section 4, we formalize the goal of boundary hiding for ciphertext streams. We give definitions for both the passive and the active adversary cases, which we call BH-CPA and BH-sfCFA. The passive case is very common in the traffic analysis literature. Here the adversary merely monitors encrypted traffic and tries to infer information from ciphertext lengths and other information such as network packet timings, but without giving away its presence by actively modifying network traffic. By hiding the ciphertext boundaries, the adversary no longer has access to fine-grained ciphertext lengths (our solution of course does not help to hide the total volume being sent). We also define boundary hiding in the active case and find out that it is much more challenging to achieve.

Denial-of-Service. Next, we focus on the very important goal of preventing fragmentation-related Denial-of-Service (DoS) attacks against the receiver. This is, to the best of our knowledge, the first formal treatment of DoS prevention as a property of encryption. For an example of such an attack, consider the SSH-CTR scheme (see [14] for a description) and the adversary who changes the length field that occupies the first 32 bits of plaintext by bit flipping in the ciphertext. If the length is maliciously increased to a very large value (say, $2^{32} - 1$, the maximum possible value for a 32-bit field), then the receiver will continue listening for ciphertext fragments awaiting message completion, until

Table 1. Stateful schemes and their security properties.

Scheme	Ref.	IND-sfCFA	BH-CPA	BH-sfCFA	DOS-sfCFA
Prefix free ($\mathcal{EC} \circ \mathcal{E}$)	(Sec. 3.3)	$+$	$\times$	$\times$	$\times$
Keyed Prefix Free (KPF)	(full version [6])	$+$	$+$	$\times$	$\times$
InterMAC	(Sec. 5)	$+$	$+$	$+$	$+$

2^{32} bytes of data have been received. Only then will SSH-CTR's MAC verification be conducted and the message rejected. The application (or user) receiving data from the SSH connection experiences this as an SSH connection hang, a form of Denial-of-Service.

We provide a security definition DOS-sfCFA for DoS attacks in Section 5 that is sufficiently flexible to capture the SSH attack and others like it. Essentially, we measure the attacker's ability to create a sequence of ciphertext fragments for which the decryption algorithm of a scheme does not output any message or failure symbol within a reasonable timeframe, measured in terms of the number of symbols submitted to a decryption oracle by the adversary.

Constructions and their security. So far, our emphasis has been on developing security models and notions. However, as we proceed, we demonstrate how each of the security notions we provide can be met in practice by efficient schemes using only standard symmetric components. These constructions are illustrative rather than definitive. Table 1 lists our main schemes for the stateful setting and their properties; for definitions and further discussions, see the referenced sections. We note that the scheme InterMAC is able to simultaneously achieve all three of our active security notions IND-sfCFA, BH-sfCFA, and DOS-sfCFA.

Further Related Work. Our fragmented approach bears more than a passing resemblance to work on on-line encryption [2, 3, 7, 9, 10, 11, 12]. However, whereas the on-line setting concerns a single continuous message and ciphertext, with each block of plaintext leading to a block of ciphertext being output during encryption (and vice-versa during decryption), our setting concerns atomic encryption (reflecting how many secure protocols operate) but allows fragmented decryption of ciphertexts. Moreover, we extensively treat the case of active adversaries, a topic that has not achieved much attention in the on-line literature, and we consider more than just confidentiality security notions. This said, our ultimate construction, InterMAC, can be seen as a kind of online scheme with a large block size.

2 Preliminaries

Since manipulating sequences of symbols (strings) in various ways will be crucial to our later exposition, we begin with some standard and not-so-standard definitions relating to strings.

Let $\mathcal{B}$ be a set and $\mathcal{B}^*$ denote the set of finite strings with symbols from $\mathcal{B}$ (including the empty string ε). Let $\mathcal{B}^+$ denote $\mathcal{B}^* \setminus \{\varepsilon\}$. Denote by $|\cdot|\colon \mathcal{B}^* \to \mathbb{Z}^+$ the length function which counts the number of symbols in a string. Typically we will have that $\mathcal{B}$ is the set of bitstrings of some fixed length n, that is $\mathcal{B} = \{0,1\}^n$, leading to the usual notation of $\{0,1\}^*$ for the set of all binary strings of finite length. In this case, if X is a string, then $|X|$ denotes its bit-length. If $\mathbf{X}$ is a vector of strings, then $|\mathbf{X}|$ denotes the number of its components. Given two strings $X, Y \in \mathcal{B}^*$, we write $X \parallel Y$ for the concatenation of the two strings. Given a sequence of strings, we define the operator $||$ that simply concatenates all the constituent strings. For example, if $X = (00)$ and $Y = (11)$, then $X \parallel Y = (0011) \in \{0,1\}^*$ and $||((00),(11)) = (0011) \in \{0,1\}^*$.

For two elements $a, b \in \mathcal{B}^*$ we call a a *prefix* of b if there exists $c \in \mathcal{B}^*$ such that $b = a \parallel c$. For two elements $a, b \in \mathcal{B}^*$ we denote with $a \star b$ the greatest common prefix of a and b and by $a \,\%\, b$ the remainder string of a with respect to b (so in particular, $a = (a \star b) \parallel (a \,\%\, b)$ and $b = (a \star b) \parallel (b \,\%\, a)$). A subset S of $\mathcal{B}^*$ is called prefix-free if for all distinct $a, b \in S$ it holds that a is not a prefix of b.

If A is finite, we can identify it with $\mathbb{Z}_{|A|}$. In the specific case of $A = \{0,1\}^n$ we use the notation $\langle\cdot\rangle_n\colon \mathbb{Z}_{2^n} \to \{0,1\}^n$ for the corresponding mapping. We extend this notion to a more general map from $\mathbb{N}$ to A^* or A^+.

3 Symmetric Encryption Supporting Fragmentation

3.1 Unified Syntax

Morphology. We extend the standard definition of symmetric encryption for the case of fragmented ciphertexts. For fragmentation to make sense, we will restrict our attention to ciphertexts that are strings, so $\mathrm{CphSp} = \mathcal{B}^*$ where e.g. $\mathcal{B} = \{0,1\}$ (bits), $\mathcal{B} = \{0,1\}^8$ (bytes), or $\mathcal{B} = \{0,1\}^{128}$ (blocks). Furthermore, we assume that the message space consists of strings, so $\mathrm{MsgSp} = \mathcal{B}^*$. The move to fragmentation results in some complications. For instance, a single ciphertext can be split up in multiple fragments or a single fragment can contain multiple ciphertexts.

Definition 1. *A symmetric encryption scheme supporting fragmentation $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ with associated* message space $\mathrm{MsgSp} = \mathcal{B}^*$, ciphertext space $\mathrm{CphSp} = \mathcal{B}^*$ *and* error messages $\mathcal{S}_\perp$ *is defined by three algorithms:*

– *The randomized key generation algorithm $\mathcal{K}$ returns a secret key K and initial states σ_0 and τ_0.*
– *The randomized or stateful (or both) encryption algorithm*

$$\mathcal{E} : \mathcal{K} \times \mathrm{MsgSp} \times \Sigma \to \mathrm{CphSp} \times \Sigma$$

takes input the secret key $K \in \mathcal{K}$, a plaintext $m \in$ MsgSp, *and optional state $\sigma \in \Sigma$, and returns a ciphertext in* CphSp *together with an updated state. For $\mathbf{m} = (m_1, \ldots, m_\ell) \in (\mathcal{B}^*)^*$ and $\mathbf{c} = (c_1, c_2, \ldots, c_\ell)$, we write*

$(\mathbf{c}, \sigma) \leftarrow \mathcal{E}_K(\mathbf{m}, \sigma_0)$ as shorthand for $(c_1, \sigma_1) \leftarrow \mathcal{E}_K(m_1, \sigma_0)$, $(c_2, \sigma_2) \leftarrow \mathcal{E}_K(m_2, \sigma_1), \ldots$ $(c_\ell, \sigma_\ell) \leftarrow \mathcal{E}_K(m_\ell, \sigma_{\ell-1})$ where $\sigma = \sigma_\ell$.

– *The deterministic and stateful decryption algorithm*

$$\mathcal{D} : \mathcal{K} \times \mathcal{B}^* \times \Sigma \to (\mathcal{B} \cup \{¶\} \cup \mathcal{S}_\perp)^* \times \Sigma$$

takes the secret key K, a ciphertext fragment $f \in \mathcal{B}^*$, and the current state τ to return the corresponding plaintext fragment m (which can be the empty string ε or an error from error space $\mathcal{S}_\perp$) and also the updated state τ. For $\mathbf{f} = (f_1, \ldots, f_\ell) \in (\mathcal{B}^*)^*$, we write $(m, \tau) \leftarrow \mathcal{D}_K(\mathbf{f}, \tau_0)$ as shorthand for $(m_1, \tau_1) \leftarrow \mathcal{D}_K(f_1, \tau_0)$, $(m_2, \tau_2) \leftarrow \mathcal{D}_K(f_2, \tau_1), \ldots$ $(m_\ell, \tau_\ell) \leftarrow \mathcal{D}_K(f_\ell, \tau_{\ell-1})$, where $m = m_1 \parallel \ldots \parallel m_\ell$ and $\tau = \tau_\ell$.

This definition requires a little unpacking. Firstly, and in contrast to the usual definitions, our decryption algorithm is stateful, mainly to model decryption algorithms that may, for example, combine data coming from multiple ciphertext fragments before outputting any plaintext.

Secondly, note that the decryption algorithm is assumed to be able to handle ciphertexts which decrypt to multiple plaintext messages, or to a mixture of plaintexts and error symbols, or possibly to nothing at all (perhaps because the input ciphertext is insufficient to enable decryption to yet output anything, giving a significant difference from the atomic setting where decryption always outputs something). We use $¶ \notin \mathcal{B} \cup \mathcal{S}_\perp$ to denote the end of plaintext messages, enabling an application making use of the decryption algorithm to parse the output uniquely into a sequence of elements of $\mathcal{B}^*$ and errors from $\mathcal{S}_\perp$. Our introduction of an explicit symbol ¶ to help delineate messages during decryption seems novel. This is not because our solution is in any way innovative, but rather because the problem does not arise in earlier works.

Thirdly, note that, when failing, the decryption algorithm can output one of possibly many error messages from the set $\mathcal{S}_\perp$. This reflects the fact that real schemes may fail in more than one way, with the different failure modes being visible to both legitimate users and adversaries. Our definition is sufficiently flexible to model schemes (such as those used in SSH and TLS) that tear down secure sessions and destroy session keys as soon as an error is detected during decryption, by having the decryption algorithm maintain an extra "abort" status flag, setting the flag once a first error is encountered, and always outputting a failure symbol once the flag is set. The definition can also handle schemes (such as those used in IPsec and DTLS) which are more tolerant of errors.

While we enforce that from a decryption of a sequence of ciphertext fragments, the corresponding message boundaries are easy to distinguish, we make no such requirement for ciphertexts. Indeed, given a sequence of ciphertext fragments, it will not be a priori clear what the constituent ciphertexts are (and in fact, in Section 4, we want to model schemes which hide these boundaries as a security goal). Looking ahead, the absence of clear ciphertext boundaries (in a sequence of fragments) will cause challenging parsing problems for our CCA definitions: in order to 'forbid' decryption of the challenge ciphertext, a prereq-

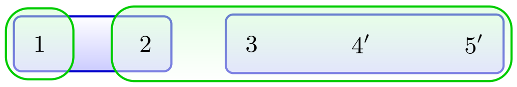

Fig. 1. Two consecutive fragments $f_1 = (1)$ and $f_2 = (234'5')$. The second fragment completes the first ciphertext $c_1 = (12)$, so we expect that to be decrypted at this point, even though ciphertext $c_2 = (345)$ in the second fragment has been modified to produce a possibly invalid ciphertext.

uisite is that this challenge ciphertext can be located accurately in the sequence of ciphertext fragments!

Correctness. If a single message is encrypted and the corresponding ciphertext is subsequently decrypted, we expect that the message is returned. When multiple messages are encrypted and the fragments correspond *exactly* to the ciphertext, again we expect to retrieve the original messages. However, we expect something stronger, namely that regardless of how we fragment the ciphertext(s), the original message(s) are returned. Moreover, we require correct decryption, even when an extra string $\mathcal{B}^*$ is added to the original (string of) ciphertexts. This forces correct decryption once a complete valid ciphertext has been received, even if what intuitively might remain in the buffer is invalid. For instance, in the situation depicted in Fig. 1 two ciphertexts $c_1 = (12)$ and $c_2 = (345)$ are produced by the encryption oracle, the adversary subsequently submits fragments $f_1 = (1)$ and $f_2 = (234'5')$ to its decryption oracle, and we still want to see the first ciphertext decrypted properly.

With this intuition in mind, we are almost ready to give our definition of correctness for a symmetric encryption scheme with fragmented ciphertexts. We first define a map $\P : (\mathcal{B}^* \cup \mathcal{S}_\perp)^* \rightarrow (\mathcal{B} \cup \{\P\} \cup \mathcal{S}_\perp)^*$ by $\P(m_1, \ldots, m_\ell) = m_1 \parallel \P \parallel \ldots \P \parallel m_\ell \parallel \P$. Note that $\P$ is injective but not surjective.

Definition 2 (Correctness Requirement). *For all* (K, σ_0, τ_0) *that can be output by* $\mathcal{K}$ *and for all* $\mathbf{m} \in \mathrm{MsgSp}^*$ *and* $\mathbf{f} \in (\mathcal{B}^*)^*$*, it holds (with probability 1) that if* $(\mathbf{c}, \sigma) \leftarrow \mathcal{E}_K(\mathbf{m}, \sigma_0)$ *and* $||(\mathbf{c})$ *prefixes* $||(\mathbf{f})$*, then* $(m', \tau) \leftarrow \mathcal{D}_K(\mathbf{f}, \tau_0)$ *satisfies* m' *is prefixed by* $\P(\mathbf{m})$*.*

Stateful versus stateless schemes. As noted in the introduction, we mainly study two subclasses of symmetric encryption schemes supporting fragmentation, *stateful*, which covers most practical encryption schemes, and *stateless beyond buffering*, an extension of standard (atomic), stateless encryption schemes that makes handling fragmented ciphertexts possible. The former case is covered by Definition 1 above. In the latter case, the decryption algorithm is still stateful, but we impose that after receiving any valid ciphertext it returns to the original state (output by key generation). More formally, we have:

```
$\mathbf{Exp}^{\mathsf{IND\text{-}sfCFA}\text{-}b}_{\mathcal{SE}}(\mathcal{A})$:
    $C \leftarrow \varepsilon, F \leftarrow \varepsilon, M \leftarrow \varepsilon$
    $\mathtt{C} \leftarrow (), \mathtt{M} \leftarrow (), i \leftarrow 0, j \leftarrow 0$
    $\mathsf{active} \leftarrow \mathtt{false}$
    $(K, \sigma, \tau) \xleftarrow{\$} \mathcal{K}$
    $b' \xleftarrow{\$} \mathcal{A}^{\mathsf{LoR}(\cdot,\cdot),\mathsf{Dec}(\cdot)}$
    return $b'$

$\mathsf{LoR}(m_0, m_1)$:
    if $|m_0| \neq |m_1|$ then return $\lightning$
    $(c, \sigma) \leftarrow \mathcal{E}_K(m_b, \sigma)$
    $i \leftarrow i + 1$
    $\mathtt{C}_i \leftarrow c, \mathtt{M}_i \leftarrow m_b$
    return $c$

$\mathsf{Dec}(f)$:
    $(m, \tau) \leftarrow \mathcal{D}_K(f, \tau)$
    $F \leftarrow F || f$ and $M \leftarrow M || m$
    if $\neg\mathsf{active}$ then
        while $C$ is a prefix of $F$ and $j < i$
            $j \leftarrow j + 1$
            $C \leftarrow C \parallel \mathtt{C}_j$
        if $F$ is prefix of $C$ then
            $m \leftarrow \varepsilon$
        else
            $\mathsf{active} \leftarrow \mathtt{true}$
            determine $m' \leftarrow \P(\mathtt{M}_1, \ldots, \mathtt{M}_{j-1})$
            extract $m \leftarrow M \,\%\, m'$
    return $m$
```

Fig. 2. The experiment defining the IND-sfCFA security notion for fragmented decryption of stateful schemes.

Definition 3. *A symmetric encryption scheme with fragmented ciphertexts is called* stateless beyond buffering *(or* sbb *for short) if it is correct (Definition 2) and satisfies the additional conditions*

1. *The initial decryption state is empty, that is for all (K, σ_0, τ_0) that can be output by $\mathcal{K}$, $\tau_0 = \varepsilon$; for simplicity's sake, we will often simply write $(K, \sigma) \xleftarrow{\$} \mathcal{K}$ for sbb schemes.*
2. *The decryption state is empty after decryption of each ciphertext obtained from encryption, i.e. for all K that can be output by $\mathcal{K}$, for all $\sigma \in \Sigma$, for all $m \in \mathrm{MsgSp}$, it holds (with probability 1) that if $(c, \sigma) \leftarrow \mathcal{E}_K(m, \sigma)$ then $(m', \tau) \leftarrow \mathcal{D}_K(c, \varepsilon)$ satisfies $\tau = \varepsilon$.*
3. *The scheme satisfies* literal decryption: *for all $K \in \mathcal{K}$ and for all $\mathbf{f} = (f_1, \ldots, f_\ell)$, when $f' = f_1 \parallel \ldots \parallel f_\ell$, then $\mathcal{D}_K(\mathbf{f}, \varepsilon) = \mathcal{D}_K(f', \varepsilon)$.*

For schemes with literal decryption we assume, without loss of generality, that the decryption algorithm only keeps a buffer ρ as state, where a buffer is understood to be a suffix of the stream of ciphertext fragments received so far. Moreover, if the scheme is sbb as well, this buffer will be emptied after each valid ciphertext. Essentially, the scheme is stateless beyond the necessary buffering (to keep track of the *current* ciphertext).

3.2 Security for Stateful Schemes

When discussing a security notion a scheme supporting fragmentation, the first thing to note is that this only makes sense in the CCA setting: if there is no decryption oracle, then whether decryption is fragmented or atomic is immaterial to the security of the scheme. In the context of fragmentation, we will replace the usual notion of chosen-ciphertext attacks by chosen-fragment attacks (CFA). Our first notion, IND-sfCFA is tailored for stateful schemes and it is inspired by

Bellare et al.'s notion of $\mathsf{IND\text{-}sfCCA}$ (for atomic schemes) from [4]. Recall that for $\mathsf{IND\text{-}sfCCA}$, an adversary has unlimited access to the decryption oracle; there are no 'prohibited' queries. Instead, to avoid trivial attacks (by the adversary simply relaying its challenge ciphertext for decryption) a syncing mechanism is used. Initially the decryption oracle is in-sync and its output (to the adversary) will be suppressed. Only when the adversary causes the decryption oracle to be out-of-sync (by deviating from the ciphertext stream output by the encryption oracle) will the purported plaintexts (or error messages) be returned.

For atomic schemes, this is relatively straightforward to define, but for schemes supporting fragmentation, some ambiguity arises. Consider again the scenario sketched in Fig. 1. The first fragment is in-sync and its output will be suppressed. In the second fragment a deviation from the challenge ciphertext stream occurs. However, *part* of the fragment is still in-sync and certainly outputting the full decryption would—mindful of the correctness requirement—reveal (part of) the plaintext (12). We will need to formalize this by officially declaring part of the fragment in-sync, and part of it out-of-sync. The ambiguity arises with regards to the boundary we should use: is sync lost already at '3' (being the first symbol of a ciphertext that is not completed properly) or only at '4' (being the first symbol of the fragment that actually deviates)?

In our definition of $\mathsf{IND\text{-}sfCFA}$ (Definition 4) we opted for the strongest interpretation, namely where synchronization is lost at the ciphertext boundary. Since this results in synchronization potentially being lost *earlier*, the decryption oracle consequently suppresses *less* of its output, making it the stronger option.

Definition 4. *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme supporting fragmentation. For an adversary $\mathcal{A}$ and a bit b, define the experiments $\mathbf{Exp}^{\mathsf{IND\text{-}sfCFA}\text{-}b}_{\mathcal{SE}}(\mathcal{A})$ as depicted in Fig. 2.*

In both experiments, first the key K is generated by $\mathcal{K}$. The adversary $\mathcal{A}$ is given access to two oracles. The first is the left-or-right encryption oracle $\mathsf{LoR}(\cdot,\cdot)$ that it can query on any pair of messages of equal length. The second oracle is the stateful decryption oracle $\mathsf{Dec}(\cdot)$ that it can query on any sequence of ciphertext fragments, but for certain sequences the output is artificially suppressed.

The adversary's goal is to output a bit b', as its guess of the challenge bit b, and the experiment returns b' as well. The $\mathsf{IND\text{-}sfCFA}$ advantage *of an adversary $\mathcal{A}$ is defined as:*

$$\mathbf{Adv}^{\mathsf{IND\text{-}sfCFA}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\Big[\mathbf{Exp}^{\mathsf{IND\text{-}sfCFA}\text{-}1}_{\mathcal{SE}}(\mathcal{A}) = 1\Big] - \Pr\Big[\mathbf{Exp}^{\mathsf{IND\text{-}sfCFA}\text{-}0}_{\mathcal{SE}}(\mathcal{A}) = 1\Big] .$$

The scheme with fragmentation $\mathcal{SE}$ is said to be indistinguishable against chosen-ciphertext-fragments attack *or* $\mathsf{IND\text{-}sfCFA}$ *secure, if for every adversary $\mathcal{A}$ with reasonable resources its* $\mathsf{IND\text{-}sfCFA}$ *advantage is small.*

Security for Stateless Schemes. In the full version [6], we define security of stateless beyond buffering encryption schemes by appropriately modifying the standard notion of indistinguishability against chosen-ciphertext attacks (IND-CCA). We also provide the details and discuss the subtleties regarding the definition.

```
Algorithm $\mathcal{K}^f$ :
  $\rho \leftarrow \varepsilon$
  $(K, \sigma, \tau) \overset{\$}{\leftarrow} \mathcal{K}^a$
  return $(K, \sigma, (\tau, \rho))$
```

```
Algorithm $\mathcal{E}^f_K(m, \sigma)$:
  $(c, \sigma') \leftarrow \mathcal{E}^a_K(m, \sigma)$
  $v \leftarrow \mathcal{EC}(c)$
  return $(v, \sigma')$
```

```
Algorithm $\mathcal{D}^f_K(f, (\tau, \rho))$:
  $w \leftarrow f$, $m' \leftarrow \varepsilon$
  $\rho \leftarrow \rho \parallel f$
  while $(w \neq \varepsilon)$
      $(w, \rho) \leftarrow \mathcal{DC}(\rho)$
      if $(w \neq \varepsilon)$ then
          $(m, \tau) \leftarrow \mathcal{D}^a_K(w, \tau)$
          $m' \leftarrow m' \parallel m \parallel \P$
  return $(m', (\tau, \rho))$
```

Fig. 3. Construction of encryption schemes supporting fragmentation.

3.3 Realizations and Non-realizations

In the full version [6], we simplify an attack by Albrecht et al. [1] to show that schemes meeting traditional notions of security in the atomic setting can fail to be secure in the fragmented setting. This establishes that our whole approach is not vacuous.

Prefix-Free Postprocessing. Next we give a simple transformation that converts any secure atomic scheme $\mathcal{SE}^a$ with MsgSp $= \mathcal{B}^*$ into a secure scheme $\mathcal{SE}^f$ supporting fragmentation. One of the challenges that has to be overcome is ensuring correct decryption of the fragmented scheme. We solve this by encoding ciphertexts (originating from $\mathcal{SE}^a$) using a prefix-free encoding scheme $\mathcal{EC}$. This allows the decrypting algorithm to correctly parse a concatenation of ciphertexts into discrete ciphertexts which it can then decrypt in an atomic fashion.

A prefix-free encoding scheme $\mathcal{EC} : \mathcal{B}^+ \rightarrow \mathcal{B}^+$ is a (deterministic) function whose image (viewed as a multiset) is prefix-free and that can be evaluated efficiently. A useful property of a prefix-free encoding scheme is that an arbitrary concatenation of encoded strings can be *uniquely* decoded, moreover this can be done instantaneously and we will assume efficiently. This property, dubbed *instantaneous decodability*, is defined below.

Definition 5. *A prefix-free encoding scheme $\mathcal{EC} : \mathcal{B}^+ \rightarrow \mathcal{B}^+$ has* instantaneous decodability *iff there exists an* efficient *deterministic algorithm $\mathcal{DC} : \mathcal{B}^* \rightarrow \mathcal{B}^* \times \mathcal{B}^*$ such that:*

1. *For all $w \in \mathcal{B}^+$ and all $s \in \mathcal{B}^*$ it holds that if $v \leftarrow \mathcal{EC}(w)$ then $(w, s) = \mathcal{DC}(v \parallel s)$.*
2. *For all $x \in \mathcal{B}^*$, if no $v \in \mathcal{EC}(\mathcal{B}^+)$ is a prefix of x then $(\varepsilon, x) = \mathcal{DC}(x)$.*

A prefix-free encoding scheme $\mathcal{EC}$ can be combined with an atomic encryption scheme with message space $\mathcal{B}^*$ to yield an encryption scheme supporting fragmentation as in Construction 6.

Construction 6 (Encrypt-then-prefix-free-encode). Let $\mathcal{SE}^a = (\mathcal{K}^a, \mathcal{E}^a, \mathcal{D}^a)$ be an atomic encryption scheme with $\mathcal{E}^a : \mathcal{B}^* \rightarrow \mathcal{B}^+$ and let $\mathcal{EC} : \mathcal{B}^+ \rightarrow \mathcal{B}^+$ be a prefix-free encoding scheme with associated (instantaneous) decoding algorithm

$\mathcal{DC}$. Then Fig. 3 defines encryption scheme supporting fragmentation $\mathcal{SE}^f = (\mathcal{K}^f, \mathcal{E}^f, \mathcal{D}^f)$.

Proposition 7. *Construction 6 provides an encryption scheme supporting fragmentation with message space* MsgSp $= \mathcal{B}^*$, *ciphertext space* CphSp $= \mathcal{B}^+$, *the same* $\mathcal{S}_\perp$ *as* $\mathcal{SE}^a$ *and it satisfies the correctness requirement given by Definition 2 (assuming* $\mathcal{SE}^a$ *itself is correct). Furthermore if* $\mathcal{D}^a$ *is stateless, then Construction 6 is stateless beyond buffering.*

Theorem 8. *If* $\mathcal{SE}^a$ *is* IND-sfCCA *secure then* $\mathcal{SE}^f$ *from Construction 6 is* IND-sfCFA *secure. More precisely, for any adversary* $\mathcal{A}_{\mathsf{sfCFA}}$ *there exists an equally efficient adversary* $\mathcal{A}_{\mathsf{sfCCA}}$ *such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}sfCFA}}_{\mathcal{SE}^f}(\mathcal{A}_{\mathsf{sfCFA}}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}sfCCA}}_{\mathcal{SE}^a}(\mathcal{A}_{\mathsf{sfCCA}})\,.$$

Theorem 9. *Let* $\mathcal{SE}^a$ *have stateless decryption. If* $\mathcal{SE}^a$ *is* IND-CCA *secure then* $\mathcal{SE}^f$ *from Construction 6 is* IND-sbbCFA *secure. More precisely, for any adversary* $\mathcal{A}_{\mathsf{sbbCFA}}$ *there exists an equally efficient adversary* $\mathcal{A}_{\mathsf{CCA}}$ *such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}sbbCFA}}_{\mathcal{SE}^f}(\mathcal{A}_{\mathsf{sbbCFA}}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathcal{SE}^a}(\mathcal{A}_{\mathsf{CCA}})\,.$$

The proofs of these results (and the definition of IND-sbbCFA security) can be found in the full version [6].

4 Boundary Hiding

In this section, we focus on formalizing the goal of boundary hiding for ciphertext streams, giving security definitions and constructions achieving these definitions. While the boundaries should be hidden to an adversary, they should of course *not* lead to decryption problems: a stream (i.e. concatenation) of ciphertexts should still lead to the correct sequence of plaintexts. The correctness requirement for an encryption scheme with fragmented decryption already ensures that everything goes well here.

Definition 10. *Let* $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be an encryption scheme supporting fragmentation. For an adversary* $\mathcal{A}$ *and a bit* b, *define experiments* $\mathbf{Exp}^{\mathsf{BH\text{-}sfCFA\text{-}}b}_{\mathcal{SE}}(\mathcal{A})$ *as depicted in Fig. 4.*

In these experiments, the adversary $\mathcal{A}$ *is given access to a special left-or-right oracle: on input two vectors of messages, either the left or the right result is returned, but with the caveat that the concatenated ciphertext is returned* only *if in both worlds the same length ciphertext is produced (but note that we do not insist that the two vectors of messages contain the same number of components). The adversary is also given access to a decryption oracle that is identical to the one provided in the* IND-sfCFA *security experiment.*

The adversary's goal is to output a bit b', *as its guess of the challenge bit* b, *and the experiment returns* b' *as well. The* BH-sfCFA advantage *of an adversary* $\mathcal{A}$ *is defined as:*

$$\mathbf{Adv}^{\mathsf{BH\text{-}sfCFA}}_{\mathcal{SE}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\mathsf{BH\text{-}sfCFA\text{-}1}}_{\mathcal{SE}}(\mathcal{A}) = 1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{BH\text{-}sfCFA\text{-}0}}_{\mathcal{SE}}(\mathcal{A}) = 1\right].$$

```
$\mathbf{Exp}_{\mathcal{SE}}^{\mathsf{BH\text{-}sfCFA}\text{-}b}(\mathcal{A})$:
  $C \leftarrow \varepsilon, F \leftarrow \varepsilon, M \leftarrow \varepsilon$
  $\mathtt{C} \leftarrow (), \mathtt{M} \leftarrow (), i \leftarrow 0, j \leftarrow 0$
  $\mathsf{active} \leftarrow \mathtt{false}$
  $(K, \sigma, \tau) \xleftarrow{\$} \mathcal{K}$
  $b' \xleftarrow{\$} \mathcal{A}^{\mathsf{LoR}(\cdot,\cdot),\mathsf{Dec}(\cdot)}$
  return $b'$

$\mathsf{LoR}(\mathbf{m}_0, \mathbf{m}_1)$:
  $\sigma_0 \leftarrow \sigma$, $\sigma_1 \leftarrow \sigma$
  $(\mathbf{c}_0, \sigma_0) \leftarrow \mathcal{E}_K(\mathbf{m}_0, \sigma_0)$
  $(\mathbf{c}_1, \sigma_1) \leftarrow \mathcal{E}_K(\mathbf{m}_1, \sigma_1)$
  $c_0 \leftarrow ||(\mathbf{c}_0)$, $c_1 \leftarrow ||(\mathbf{c}_1)$
  if $|c_0| \neq |c_1|$ then return $\lightning$
  $\sigma \leftarrow \sigma_b$
  for $\iota = 1$ to $\iota = |\mathbf{c}_b|$
    $i \leftarrow i + 1$
    $\mathtt{C}_i \leftarrow \mathbf{c}_b(\iota), \mathtt{M}_i \leftarrow \mathbf{m}_b(\iota)$
  return $c_b$

$\mathsf{Dec}(f)$:
  $(m, \tau) \leftarrow \mathcal{D}_K(f, \tau)$
  $F \leftarrow F||f$ and $M \leftarrow M||m$
  if $\neg\mathsf{active}$ then
    while $C$ is a prefix of $F$ and $j < i$
      $j \leftarrow j + 1$
      $C \leftarrow C \,||\, \mathtt{C}_j$
    if $F$ is prefix of $C$ then
      $m \leftarrow \varepsilon$
    else
      $\mathsf{active} \leftarrow \mathtt{true}$
      determine $m' \leftarrow \P(\mathtt{M}_1, \ldots, \mathtt{M}_{j-1})$
      extract $m \leftarrow M \,\%\, m'$
  return $m$
```

Fig. 4. Experiment $\mathbf{Exp}_{\mathcal{SE}}^{\mathsf{BH\text{-}sfCFA}\text{-}b}(\mathcal{A})$ for defining boundary hiding security for stateful schemes and an active adversary.

We say that $\mathcal{SE}$ is boundary-hiding against chosen-ciphertext-fragments attack *or* BH-sfCFA *secure, if for every adversary $\mathcal{A}$ with reasonable resources its* BH-sfCFA *advantage is small.*

Boundary-hiding notions for the case of passive adversaries can be obtained simply by removing the adversary's access to the relevant decryption oracle. In this case, we refer to BH-CPA security (this notion is implied by the notion IND\$-CPA as introduced by Rogaway [17], see the full version [6]). The notion of boundary-hiding security for sbb schemes for active attacks BH-sbbCFA can also be developed, by replacing the decryption oracle Dec in Fig. 4 by the decryption oracle Dec from the corresponding sbb security game and by appropriately modifying how ciphertexts generated by queries to the encryption oracle are tracked. The details are in the full version [6].

Constructions. In order to achieve BH-CPA security we extend Construction 6 by using *keyed* encoding schemes. The use of a key in the encoding scheme enables the encryption algorithm to disguise the ciphertext boundaries from a passive adversary. Details of this construction, dubbed KPF (Keyed Prefix Free), can be found in the full version [6]. Achieving BH-sfCFA requires more work. We show that this can be done at the same time as achieving DoS security in the next section. In the sbb setting, however, we will only achieve BH-CPA security (we show this and discuss the difficulty of meeting BH-sbbCFA security in the next section as well). It remains an open problem to design a practical BH-sbbCFA secure sbb scheme.

```
$\mathbf{Exp}_{\mathcal{SE}}^{N\text{-DOS-sfCFA}}(\mathcal{A})$:
    $C \leftarrow \varepsilon, F \leftarrow \varepsilon, M \leftarrow \varepsilon$
    $\mathtt{C} \leftarrow (), \mathtt{M} \leftarrow (), i \leftarrow 0, j \leftarrow 0$
    $\mathsf{active} \leftarrow \mathtt{false}$
    $(K, \sigma, \tau) \xleftarrow{\$} \mathcal{K}$
    run $\mathcal{A}^{\mathsf{Enc}(\cdot),\mathsf{Dec}(\cdot)}$
    return $1$

$\mathsf{Enc}(m)$:
    $(c, \sigma) \leftarrow \mathcal{E}_K(m, \sigma)$
    $i \leftarrow i + 1$
    $\mathtt{C}_i \leftarrow c, \mathtt{M}_i \leftarrow m$
    return $c$

$\mathsf{Dec}(f)$:
    $(m, \tau) \leftarrow \mathcal{D}_K(f, \tau)$
    $F \leftarrow F || f$ and $M \leftarrow M || m$
    if $\neg\mathsf{active}$ then
        while $C$ is a prefix of $F$ and $j < i$
            $j \leftarrow j + 1$
            $C \leftarrow C \,\|\, \mathtt{C}_j$
        if $F$ is not a prefix of $C$ then
            $\mathsf{active} \leftarrow \mathtt{true}$
            $m' \leftarrow \P(\mathtt{M}_1, \ldots, \mathtt{M}_{j-1})$
            $m \leftarrow M \,\%\, m'$
    if $\mathsf{active}$ then
        if $m \neq \varepsilon$ then
            exit Exp with $0$
        else if $|F \,\%\, C| \geq N$ then
            exit Exp with $1$
    return $m$
```

Fig. 5. The experiment defining the N-DOS-sfCFA security notion for Denial of Service attack against stateful schemes.

5 Denial-of-Service Attacks

In this section we study fragmentation-related Denial-of-Service (DoS) attacks. In the introduction we mentioned an example of a fragmentation-related attack that constitutes DoS. Here we provide formal definitions that are general enough to capture all such attacks. We focus on the stateful setting.

Definition 11. *Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a stateful encryption scheme with fragmented ciphertexts. For an adversary $\mathcal{A}$ and $N \in \mathbb{N}$ define the experiment $\mathbf{Exp}_{\mathcal{SE}}^{N-\mathsf{DOS\text{-}sfCFA}}(\mathcal{A})$ as in Fig. 5. In the experiment, $\mathcal{A}$ is given access to two oracles. The first is a regular encryption oracle $\mathsf{Enc}(\cdot)$ that it can query on any message in the message space. The second oracle is a special stateful decryption oracle $\mathsf{Dec}(\cdot)$ that it can query on any string treated as a ciphertext fragment. The adversary's goal is to submit to the special decryption oracle $\mathsf{Dec}(\cdot)$ a fragment or a sequence of fragments of length at least N, which is not a valid replay of legitimate ciphertexts and such that $\mathsf{Dec}(\cdot)$ does not return a non-empty message m or a failure symbol from $\mathcal{S}_\perp$. In this case the oracle exits the experiment with a value 1. In other out-of-sync cases the oracle exits with 0. When decryption is still in-sync, the oracle just returns the correct decryption m. The N-$\mathsf{DOS\text{-}sfCFA}$ advantage of $\mathcal{A}$ is defined to be:*

$$\mathbf{Adv}_{\mathcal{SE}}^{N\text{-}\mathsf{DOS\text{-}sfCFA}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathcal{SE}}^{N\text{-}\mathsf{DOS\text{-}sfCFA}}(\mathcal{A}) = 1\right].$$

The scheme $\mathcal{SE}$ is said to be N-DOS-sfCFA secure *if for every legitimate adversary $\mathcal{A}$ with reasonable resources, its N-DOS-sfCFA advantage is small.*

Note that, to win the above game, the adversary need not make his attack in the first out-of-sync query to $\mathsf{Dec}(\cdot)$. Also note that in order to win the adversary must submit at least N symbols after the longest common prefix with a valid ciphertext stream obtained from the encryption oracle without provoking any output from the decryption oracle.

The parameter N in the definition above measures the shortest fragment length below which a DoS attack cannot be prevented by a scheme; since all reasonable schemes that meet our other security notions must do some degree of buffering before outputting any message symbols, we cannot hope to make N as small as we please. Our objective then, when designing a scheme, is to make the scheme N-$\mathsf{DOS\text{-}sfCFA}$ secure for N as small as possible.

We develop a similar definition for the sbb setting in the full version [6].

InterMAC: construction in the stateful case and its security. Our idea for DoS prevention in the stateful setting is to break the ciphertexts into equal-sized segments and authenticate all of them. We could use this idea to modify an $\mathsf{IND\text{-}sfCFA}$ scheme, but we propose a more efficient construction that uses an $\mathsf{IND\text{-}CPA}$ (possibly stateless) scheme and a $\mathsf{SUF\text{-}CMA}$ MAC. (We defer standard definitions for syntax and security of MACs to the full version [6].) In our construction, the sender and receiver keep a state which contains a message and a segment number. The encryption algorithm MACs this state together with the encryption of the segment, but the state does not have to be transmitted, as the receiver maintains it for himself. Each segment uses a bit flag to indicate the last segment in a message. We now provide the details.

Construction 12 (InterMAC). Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be an encryption scheme with associated message space MsgSp_e and let $\mathcal{MAC} = (\mathcal{K}_t, \mathcal{T}, \mathcal{V})$ be a message authentication code with associated message space MsgSp_t. Let $N \in \mathbb{N}$ be a DoS parameter. We assume that $\mathrm{MsgSp}_t = \{0,1\}^*$ and, for simplicity, that $\mathrm{MsgSp}_e = \{\{0,1\}^{N-el-1-tl}\}^*$ where $\mathcal{T}$ always outputs tags of fixed length tl and $\mathcal{E}$ always produces ciphertexts which are el bits longer than the messages. This restriction on the message space can be relaxed by introducing an appropriate padding scheme (such as abit padding, as analysed in [15]) but we omit this detail for simplicity. Define a new stateful encryption scheme with fragmentation $\mathcal{SE}^f = (\mathcal{K}^f, \mathcal{E}^f, \mathcal{D}^f)$ as in Figure 6.

It is not hard to check that the scheme is correct.

The proofs of the following two theorems are in the full version [6].

Theorem 13. *If $\mathcal{MAC}$ is $\mathsf{SUF\text{-}CMA}$, then $\mathcal{SE}^f$ constructed as per Construction 12 is N-$\mathsf{DOS\text{-}sfCFA}$ secure. More precisely, for any adversary $\mathcal{A}$ there exists an equally efficient adversary $\mathcal{A}'$ so that*

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{MAC}}(\mathcal{A}') \geq \mathbf{Adv}^{N\text{-}\mathsf{DOS\text{-}sfCFA}}_{\mathcal{SE}^f}(\mathcal{A}).$$

Theorem 14. *If $\mathcal{SE}$ is $\mathsf{IND\$\text{-}CPA}$[1] and $\mathcal{MAC}$ is $\mathsf{SUF\text{-}CMA}$ and PRF secure, then $\mathcal{SE}^f$ constructed as per Construction 12 is $\mathsf{BH\text{-}sfCFA}$ and $\mathsf{IND\text{-}sfCFA}$ secure.*

We provide the concrete security statements in [6].

[1] This notion captures indistinguishability of ciphertexts from random strings and is introduced by Rogaway [17]. We recall the formal definition in [6].

```
Algorithm $\mathcal{E}^f_{K_e\|K_t,\sigma}(m;\sigma)$:
    $\sigma \leftarrow \sigma + 1$
    parse $m$ as $m_1 \| \ldots \| m_\ell$
        where for each $1 \leq i \leq \ell$,
        $|m_i| = N - el - 1 - tl$
    for $j = 1$ to $\ell$ do
        if $j = \ell$ then
            $bm_j \leftarrow 1 \| m_j$
        else $bm_j \leftarrow 0 \| m_j$
        $c_j \leftarrow \mathcal{E}_{K_e}(bm_j)$
        $t_j \leftarrow \mathcal{T}_{K_t}(\sigma \| j \| c_j)$
    $c \leftarrow c_1 \| t_1 \ldots \| c_\ell \| t_\ell$
    return $(c, \sigma)$
```

```
Algorithm $\mathcal{D}^f_{K_e\|K_t}(f, \tau)$:
    parse $\tau$ as $(i_m, i_s, bad, m, F)$
    if $|F \| f| < N$ then
        $F \leftarrow F \| f$
        return $(\varepsilon, (i_m, i_s, bad, m, F))$
    else parse $F \| f$ as $c_1 \| t_1 \| \ldots \| c_\ell \| t_\ell \| s$
        where $|s| < N$ and for each $1 \leq i \leq \ell$,
        $|c_i \| t_i| = N$ and $|t_i| = tl$
    for $j = 1$ to $\ell$ do
        $i_s \leftarrow i_s + 1$
        if $bad = 1$ then
            output $\perp$
        else if $\mathcal{V}_{K_t}(i_m \| i_s \| c_j, t_j) = 0$ then
            $bad \leftarrow 1$ ; output $\perp$
        else
            $bm_j \leftarrow \mathcal{D}_{K_e}(c_j)$ ; parse $bm_j$ as $b \| m_j$
            $m \leftarrow m \| m_j$
            if $b = 1$ then
                $i_m \leftarrow i_m + 1$; $i_s \leftarrow 0$
                if $bad = 0$ then
                    output $(m¶)$ ; $m \leftarrow \varepsilon$
    return$(\varepsilon, (i_m, i_s, bad, m, s))$
```

Fig. 6. The stateful construction $\mathcal{SE}^f$. Key generation $\mathcal{K}^f$ picks $K_e \xleftarrow{\$} \mathcal{K}_e, K_t \xleftarrow{\$} \mathcal{K}_t$, sets $\sigma \leftarrow 0$ and $\tau = (i_m, i_s, bad, m, F) \leftarrow (0, 0, 0, \varepsilon, \varepsilon)$, and returns (c, σ, τ).

Construction in the sbb case and its security. In the full version [6] we provide an analogous scheme for the sbb setting. We also use the idea of authenticating the ciphertext segments, however, the solution becomes more complex as we cannot keep message and segment numbers as state and it is not efficient to keep them as part of the segments. To prevent the re-ordering attacks we need to authenticate the previous segments as well. To improve efficiency we only authenticate the tags from the previous segments. The security results we get are similar, but the sbb scheme does not provide BH-sbbCFA security.

The reason for the difficulty in achieving BH-sbbCFA security in the sbb setting is as follows. An adversary can always query a valid ciphertext to the stateful decryption oracle fragment by fragment and flip the last bit at the end. Observing when the decryption algorithm returns $\perp$ gives the adversary information about the ciphertext boundary. Prohibiting the decryption algorithm to ever return $\perp$ is not very practical and is subject to DoS attacks. It is an open question to provide a practical scheme with both BH-sbbCFA and DOS-sbbCFA security.

6 Conclusions

In this paper, we have initiated the formal study of fragmentation attacks against symmetric encryption schemes. We also developed security models to formalise the additional desirable properties of ciphertext boundary-hiding and robustness against Denial-of-Service (DoS) attacks for schemes in this setting. We illustrated the utility of each of our models via efficient constructions for schemes using only standard cryptographic components. This work raises many interesting open questions, amongst which we list:

- We have focussed on confidentiality notions here, and suitable integrity notions remain to be developed. Can such notions then be combined to provide more general notions of security as seen in authenticated encryption?
- Some of our constructions build fragmented schemes from atomic schemes. What general relationships are there between schemes in the two settings?
- In the sbb case, the properties of DoS resistance and ciphertext boundary hiding appear to be in opposition to one another. Can this be formally proven? Is there a fundamental reason (beyond the ability to keep state) why this does not seem to arise in the stateful setting?

References

[1] M.R. Albrecht, K.G. Paterson, and G.J. Watson. Plaintext recovery attacks against SSH. In *IEEE Symposium on Security and Privacy*, pages 16–26. IEEE Computer Society, 2009.

[2] G.V. Bard. A challenging but feasible blockwise-adaptive chosen-plaintext attack on SSL. In M. Malek, E. Fernandez-Medina and J. Hernando (eds.), SECRYPT, pages 99–109. INSTICC Press, 2006.

[3] G.V. Bard. Blockwise-adaptive chosen-plaintext attack and online modes of encryption. In S.D. Galbraith (ed.), IMA Int. Conf., volume 4887 of *Lecture Notes in Computer Science*, pages 129–151. Springer, 2007.

[4] M. Bellare, T. Kohno, and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the encode-then-encrypt-and-MAC paradigm. *ACM Transactions on Information and Systems Security*, 7(2):206–241, 2004.

[5] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto (ed.), *ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.

[6] A. Boldyreva, J. P. Degabriele, K. G. Paterson and M. Stam. Security of symmetric encryption in the presence of ciphertext fragmentation. Full version of this paper. Available from *Cryptology ePrint Archive* http://eprint.iacr.org, 2012.

[7] A. Boldyreva and N. Taesombut. Online encryption schemes: New security notions and constructions. In T. Okamoto (ed.), CT-RSA, volume 2964 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2004.

[8] J.P. Degabriele and K.G. Paterson. On the (in)security of IPsec in MAC-then-Encrypt configurations. In E. Al-Shaer, A.D. Keromytis and V. Shmatikov (eds.), *ACM Conference on Computer and Communications Security*, pages 493–504, ACM, 2010.

[9] P.-A. Fouque, A. Joux, G. Martinet and F. Valette. Authenticated on-line encryption. In M. Matsui and R.J. Zuccherato (eds.), SAC, volume 3006 of *Lecture Notes in Computer Science*, pages 145–159. Springer, 2003.

[10] P.-A. Fouque, A. Joux and G. Poupard. Blockwise adversarial model for on-line ciphers and symmetric encryption schemes. In H. Handschuh and M.A. Hasan (eds.), SAC, volume 3357 of *Lecture Notes in Computer Science*, pages 212–226. Springer, 2004.

[11] P.-A. Fouque, G. Martinet and G. Poupard. Practical symmetric on-line encryption. In T. Johansson, ed., FSE, volume 2887 of *Lecture Notes in Computer Science*, pages 362–375. Springer, 2003.

[12] A. Joux and G. Martinet and F. Valette Blockwise-Adaptive Attackers: Revisiting the (In)Security of Some Provably Secure Encryption Models: CBC, GEM, IACBC. In M. Yung (ed.), *CRYPTO 2001*, volume 2442 of *Lecture Notes in Computer Science*, pages 17–30, Springer, 2002.

[13] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In J. Kilian (ed.), *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer, 2001.

[14] K.G. Paterson and G.J. Watson. Plaintext-dependent decryption: A formal security treatment of SSH-CTR. In H. Gilbert (ed.), *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 345–361. Springer, 2010.

[15] K.G. Paterson and G.J. Watson. Immunising CBC Mode Against Padding Oracle Attacks: A Formal Security Treatment. In R. Ostrovsky, R. De Prisco and I. Visconti (eds.), *SCN 2008*, volume 5229 of *Lecture Notes in Computer Science*, pages 340–357, Springer, 2008.

[16] K.G. Paterson, T.E. Shrimpton and T. Ristenpart. Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol. In D.H. Lee and X. Wang (eds.), *ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 372–389, Springer, 2011.

[17] P. Rogaway. Nonce-based symmetric encryption. In B. Roy and W. Meier (eds.), *FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 348–359, Springer 2004.

[18] C. Tezcan and S. Vaudenay. On Hiding a Plaintext Length by Preencryption. In J. Lopez, G. Tsudik (eds.), *ACNS 2011*, volume 6715 of *Lecture Notes in Computer Science*, pages 345–358, Springer, 2011.