Automated Verification of Block Cipher Modes of Operation, an Improved Method

Martin Gagné[1], Pascal Lafourcade[1],
Yassine Lakhnech[1], and Reihaneh Safavi-Naini[2]

[1] Université Grenoble 1, CNRS, Verimag, France
[2] Department of Computer Science, University of Calgary, Canada

Abstract. In this paper, we improve on a previous result by Gagné et al. [9] for automatically proving the semantic security of symmetric modes of operation for block ciphers. We present a richer assertion language that uses more flexible invariants, and a more complete set of rules for establishing the invariants. In addition, all our invariants are given a meaningful semantic definition, whereas some invariants of the previous result relied on more ad hoc definitions. Our method can be used to verify the semantic security of all the encryption modes that could be proven secure in [9], in addition to other modes, such as Propagating Cipher-Block Chaining (PCBC).

1 Introduction

Block ciphers are among the most basic building blocks in cryptography. They can be used to construct primitives as varied as message authentication codes, hash functions and, their main application, symmetric encryption. Block ciphers are deterministic, and have fixed-size input and output, so protocols, called modes of operation, are required to encrypt messages of arbitrary length. The security of these modes of operation is then proven by reduction from the security of the mode of operation to some security property of the block cipher.

Automated verification tools can help increase our confidence in the security of these modes of operation by providing an independent argument for their security. Gagné et al. [9] first initiated the study of automatic verification techniques for symmetric modes of operation. They presented an assertion language, invariants and rules for a Hoare logic which can be used to verify the security of most of the traditional modes of operation. However, due to the rather ad hoc nature of the description of certain invariants, and to the restrictiveness of their rule set, the resulting automated verifier was limited and its results could sometimes depend on the order in which the commands of the mode of operation were written.

Contributions: We improve on the result of Gagné et al. [9] by presenting a Hoare logic with a richer assertion language and invariants, which allow us to verify more modes of operation. For example, our new logic is able to verify the security of Propagating Cipher-Block Chaining (PCBC) – an encryption mode that was introduced for Kerberos version 4 – while [9] could not.

The programming language and assertion language are essentially the same as [9], but our invariants are much more precise. We use only three predicates: one that states that the value of a variable is indistinguishable from a random value, one that states that the block cipher has never been computed at the value of a variable, and one that keeps track of the most recent value of a counter. Our predicates are also much more satisfying than those in [9] since they can all be described using a clear semantic definition, whereas some predicates in [9] were rather ad hoc, particularly when it came to the predicate used to keep track of counters.

Using our logic as a set of rules for propagating the invariants though the code of each mode of operation, we can verify the semantic security of all the encryption modes which could be shown secure in [9], together with other modes, such as PCBC.

Related Work: Many new modes of operation have been developped in the last decade (IACBC, IAPM [17], XCB [20], TMAC [16,18], HCTR [4], HCH [6], EMU [13], EMU* [10], PEP [5], OMAC [14,15], TET [11], CMC [12], XEX [21], TAE, TCH, TBC [19,22] to name only a few). These new modes of operation often offer new security functionalities that early modes did not possess, but these properties come at the cost of increased complexity of the mode of operation, and increased complexity of the proof of security. We believe that automated verification of these modes would greatly increase our confidence in those arguments.

An extensive discussion on different security notions for symmetric encryption and a proof of the CBC mode of encryption is presented in [3]. They also present a security proof of the CBC mode of operation through a reduction to the security of the block cipher.

An automatic method for proving the semantic security for asymmetric encryption schemes in the random oracle model was presented in [7]. A similar method is used in [9] to verify the security of symmetric encryption modes. Our work here is a continuation of these efforts.

Other works in automated verification of cryptographic protocols include [1], which presents a new logic for reasoning about cryptographic primitives, and uses this logic to formally analyze the security of the signature scheme PSS, and [2], which provides a machine-checked proof of OAEP.

We refer the reader to our technical report [8] for the complete details on this result, definitions, rule set and proofs. We will focus here on the new semantics and our new definition for the invariants, and explain how they improve our verifier.

2 Definitions

The encryption modes verified using our method must be written using the language described by the grammar of Figure 1, where:

$$\begin{aligned} \mathrm{c} ::= x \overset{\$}{\leftarrow} \mathcal{U} \mid x := \mathcal{E}(y) \mid x := \mathcal{E}^{-1}(y) \mid x := y \oplus z \mid x := y \| z \mid x := y[n,m] \mid \\ \mid x := y + 1 \mid c_1; c_2 \end{aligned}$$

Fig. 1. Language grammar

- $x \overset{\$}{\leftarrow} \mathcal{U}$ denotes uniform sampling of a value and assigning it to x.
- $x := \mathcal{E}(y)$ denotes application of the block cipher $\mathcal{E}$ to the value of y and assigning the result to x.
- Similarly for $x := \mathcal{E}^{-1}(y)$, where $\mathcal{E}^{-1}$ denotes the inverse function of $\mathcal{E}$.
- $x := y \oplus z$ denotes application of the exclusive-or operator to the values of y and z and assigning the result to x.
- $x := y||z$ represents the concatenation of the values of y and z.
- $x := y[n, m]$ assigns to x the bits at positions between n and m in the bit-string value of y. I.e., for a bit-string $bs = b_1 \dots b_k$, where the b_i's are bits, $bs[n, m]$ denotes the bits-string $b_n \dots b_m$[1]. Then, $x := y[n, m]$ assigns $bs[n, m]$ to x, where bs is the value of y. Here, n and m are polynomials in the security parameter η.
- $x := y + 1$ increments by one the value of y and assigns the result to x. The operation is carried modulo 2^η
- $c_1; c_2$ is the sequential composition of c_1 and c_2

We can now formally define a mode of encryption as follows:

Definition 1 (Generic Encryption Mode). *A generic encryption mode M is represented by $\mathcal{E}_M(m_1|\dots|m_i, c_0|\dots|c_i)$: **var** $\boldsymbol{x_i}; \mathsf{c}_i$, where $\boldsymbol{x_i}$ is the set of variables used in c_i, all commands of c_i are built using the grammar described in Figure 1, each m_j is a message blocks, and each c_j is a cipher block, both of size n according to the input length of the block cipher $\mathcal{E}$.*

2.1 Semantics

A state in our semantics consists of an assignment of value to each variable used in the program (a set denoted by Var) in addition to a list $\mathcal{L}_\mathcal{E}$ which keeps track of all the values on which the block cipher has been computed. We also use a table $\mathcal{T}$ which keeps track of all the variables which are used as counters[2]. Our new semantics are described in Table 1.

2.2 New Invariants

We introduce a new invariant, $\mathsf{Icounter}()$ which is defined using our new table $\mathcal{T}$, and make small modifications to the other invariants[3]. Of particular interest: weaugment the invariant Indis so that it is now possible to indicate that the value

[1] Notice that $bs[n, m] = \epsilon$, when $m < n$ and $bs[n, m] = bs[n, k]$, when $m > k$.

[2] We drop the sets F and C, which were an ad hoc attempt at keeping track of fresh variables and counters.

[3] We do away with invariants F and $RCounter$, which were defined using the also-removed sets F and C from the semantics.

Table 1. The semantics of the programming language

$$
\begin{array}{l}
[\![x \overset{r}{\leftarrow} \mathcal{U}]\!](S,\mathcal{E}) = [u \overset{r}{\leftarrow} \mathcal{U} : (S\{x \mapsto u, \mathcal{T} \mapsto \mathcal{T} \cup \{\mathcal{T}_x\}, \mathcal{E})] \\
[\![x := \mathcal{E}(y)]\!](S,\mathcal{E}) = \\
\qquad \begin{cases} \delta(S\{x \mapsto v, \mathcal{T}, \mathcal{E}) \text{ if } (S(y), v) \in \mathcal{L}_\mathcal{E} \\ \delta(S\{x \mapsto v, \mathcal{T} \mapsto \mathcal{T} \cup \{\mathcal{T}_x\}, \mathcal{L}_\mathcal{E} \mapsto S(\mathcal{L}_\mathcal{E}) \cdot (S(y), v)\}, \mathcal{E}) \\ \qquad\qquad\qquad\qquad\qquad \text{if } (S(y), v) \notin \mathcal{L}_\mathcal{E} \text{ and } v = \mathcal{E}(S(y)) \end{cases} \\
[\![x := y \oplus z]\!](S,\mathcal{E}) = \delta(S\{x \mapsto S(y) \oplus S(z), \mathcal{T}, \mathcal{E}) \\
[\![x := y || z]\!](S,\mathcal{E}) = \delta(S\{x \mapsto S(y) || S(z), \mathcal{T}, \mathcal{E}) \\
[\![x := y[n,m]]\!](S,\mathcal{E}) = \delta(S\{x \mapsto S(y)[n,m], \mathcal{T}, \mathcal{E}) \\
[\![x := y + 1]\!](S,\mathcal{E}) = \\
\qquad \begin{cases} \delta(S\{x \mapsto S(y)+1, \mathcal{T} \mapsto \mathcal{T} \cup \{\mathcal{T}_z \mapsto \mathcal{T}_z[i+1] = x\}, \mathcal{E}) \text{ if } y = \mathcal{T}_z[i] \wedge \mathcal{T}_z[i+1] = \bot \\ \delta(S\{x \mapsto S(y)+1, \mathcal{T}, \mathcal{E}) \text{ otherwise} \end{cases} \\
[\![c_1; c_2]\!] = [\![c_2]\!] \circ [\![c_1]\!]
\end{array}
$$

of a variable is indistinguishable from a random value when given all the values in $\mathcal{L}_\mathcal{E}$. This small modification is crucial to the Lemma below, and is one of the main reasons for the improved capacity of our automated prover.

lcounter$(x; V)$**:** means that x is the most recent value of a counter that started at a random value, and that the set V contains all the variables with previous values of the counter.

E$(\mathcal{E}; x; V)$**:** means the probability that the value of x is neither in $\mathcal{L}_\mathcal{E}$ nor in V is negligible.

Indis$(\nu x; V)$**:** means that no adversary has non-negligible probability to distinguish the value of x from a random value, when he is given the values of the variables in V. In addition to variables in Var, the set V can contain a special variable $\mathcal{L}_\mathcal{E}$, in which case the invariant means that no adversary has non-negligible probability to distinguish whether he is given results of computations performed using the value of x or a random value, when he is given the values of the variables in V and $\mathcal{L}_\mathcal{E}$.

More formally, for each invariant ψ, we define that a distribution X satisfies ψ, denoted $X \models \psi$ as follows:

– When $\mathcal{L}_\mathcal{E} \notin V$, $X \models \mathsf{Indis}(\nu x; V)$ iff $[(S,\mathcal{E}) \overset{r}{\leftarrow} X : (S(x,V), \mathcal{E})] \sim [(S,\mathcal{E}) \overset{r}{\leftarrow} X; u \overset{r}{\leftarrow} \mathcal{U}; S' = S\{x \mapsto u\} : (S'(x,V), \mathcal{E})]$
– When $\mathcal{L}_\mathcal{E} \in V$, $X \models \mathsf{Indis}(\nu x; V)$ iff $[(S,\mathcal{E}) \overset{r}{\leftarrow} X : (S(x, V \cup \mathcal{L}_\mathcal{E}.dom), \mathcal{E})] \sim [(S,\mathcal{E}) \overset{r}{\leftarrow} X; u \overset{r}{\leftarrow} \mathcal{U}; S' = S\{x \mapsto u\} : (S'(x, V \cup \mathcal{L}_\mathcal{E}.dom), \mathcal{E})]$
– $X \models \mathsf{E}(\mathcal{E}; x; V)$ iff $\mathsf{Pr}[(S,\mathcal{E}) \overset{r}{\leftarrow} X : S(x) \in S(\mathcal{L}_\mathcal{E}).\mathsf{dom} \cup S(V)]$ is negligible.
– $X \models \mathsf{lcounter}(x; V)$ iff $\mathsf{Indis}(x; \mathsf{Var} \setminus V)$ and $V = \mathcal{T}(x)$.

where $\mathcal{L}_\mathcal{E}.dom = \{v_1 \mid (v_1, v_2) \in \mathcal{L}_\mathcal{E}\}$ and $\mathcal{T}(x) = \{x \in \mathsf{Var} \mid \exists i, j \in \mathbb{N}$ and $y \in \mathsf{Var}$ such that $\mathcal{T}_y[i] = x$ and $\mathcal{T}_y[j] = y\}$.

Notation: For a set V and a variable, we write V, x as a shorthand for $V \cup \{x\}$ and $V - x$ as a shorthand for $V \setminus \{x\}$. We denote by Var^* the set $\mathsf{Var} \cup \mathcal{L}_\mathcal{E}$ and use $\mathsf{Indis}(\nu x)$ as a shorthand for $\mathsf{Indis}(\nu x; \mathsf{Var}^*)$.

The relation between these invariants are described in the following Lemma.

Lemma 1. *For any set $V \subset \mathsf{Var}$ and variables x, y with $x \neq y$, we have*

1. $\mathsf{Indis}(\nu x; V \cup \mathcal{L}_{\mathcal{E}}) \Rightarrow \mathsf{E}(\mathcal{E}; x; V \setminus \{x\})$
2. $\mathsf{Icounter}(x; V) \Rightarrow \mathsf{Indis}(x; \mathsf{Var} \setminus V)$
3. $\mathsf{E}(\mathcal{E}; x; V) \wedge \mathsf{Indis}(\nu x; \{y\}) \Rightarrow \mathsf{E}(\mathcal{E}; x; V, y)$

The first line of this Lemma is particularly important, because it links the Indis invariant to the E invariant. This is of interest because the invariant Indis is quite a bit easier to deal with than the invariant E alone, so this enables us to infer that quite a few more variables have never been queried to the block cipher than in our previous paper (however, we have to be careful to handle correctly rules for commands that add elements to $\mathcal{L}_{\mathcal{E}}$, but this is also done relatively easily). As a result of this, it is possible, for example, to 'pass along' the invariant E to multiple values in a chain of Xor operations – whereas it was only possible to pass it once in our previous paper – which is what makes it now possible to prove the security of the PCBC mode of operation.

2.3 Encryption Security

We prove the modes of encryption secure in the ideal cipher model. That is, we assume that the block cipher is a pseudo-random function.[4] This is a standard assumption for proving the security of any block-cipher-based scheme.

The semantic security for a mode of encryption is defined as follows.

Definition 2. *Let $\mathcal{E}_M(m_1|\ldots|m_i, c_0|\ldots|c_i)$: **var** $\boldsymbol{x_i}$; c_i be a generic encryption mode. $A = (A_1, A_2)$ be an adversary and $X \in \textsc{Dist}(\Gamma, \mathcal{E})$. For $\eta \in \mathbb{N}$, let*

$$\begin{aligned}
\mathsf{Adv}^{ind-CPA}_{A,M}&(\eta, X)\\
&= 2 * \mathsf{Pr}[(S, \mathcal{E}) \xleftarrow{r} X;\\
&\quad (x_0, x_1, p, s) \xleftarrow{r} A_1^{\mathcal{O}_1}(\eta); b \xleftarrow{r} \{0,1\};\\
&\quad S' \xleftarrow{r} [\![c_p]\!](S\{m_1|\ldots|m_p \mapsto x_b\}, \mathcal{E}) :\\
&\quad A_2^{\mathcal{O}_2}(x_0, x_1, s, S'(c_0|\ldots|c_p)) = b] - 1
\end{aligned}$$

where $\mathcal{O}_1 = \mathcal{O}_2$ are oracles that take a pair (m, j) as input, where m is a string and j is the block length of m, and answers using the $j^t h$ algorithm in $\mathcal{E}_M$. A_1 outputs x_0, x_1 such that $|x_0| = |x_1|$ and are composed of p blocks. The mode of operation M is semantically (IND-CPA) secure if $\mathsf{Adv}^{ind-CPA}_{A,M}(\eta, X)$ is negligible for any constructible distribution ensemble X and polynomial-time adversary A.

Our method verifies the security of an encryption scheme by proving that the ciphertext is indistinguishable from random bits. It is a classical result that this implies semantic security. More precisely:

[4] While block ciphers are really families of permutations, it is well known that pseudo-random permutations are indistinguishable from pseudo-random functions if the block size is large enough.

Proposition 1. *Let $\mathcal{E}_M(m_1|\dots|m_i, c_0|\dots|c_i)$: **var** $\boldsymbol{x_i}$; c_i be a generic encryption mode. If, after execution of c_i the invariant $\bigwedge_{j=0}^{i} \mathit{Indis}(\nu c_j, IO)$ holds, where $IO = \{m_1, \dots, m_i, c_0 \dots, c_i\}$, then the encryption mode is semantically secure.*

3 Hoare Logic Rules

In the following, the notation $\{\varphi\}\mathsf{c}\{\varphi'\}$ means that execution of command c in any distribution that satisfies φ leads to a distribution that satisfies φ'. Using Hoare logic terminology, this means that the triple $\{\varphi\}\mathsf{c}\{\varphi'\}$ is valid. We group rules together according to their corresponding commands. We do not provide rules for the commands $x := \mathcal{E}^{-1}(y)$ or $x := y[n, m]$ since those commands are only used during decryption.

In all the rules below, unless indicated otherwise, we assume that $t \notin \{x, y, z\}$ and $x \notin \{y, z\}$. In addition, for all rules involving the invariant Indis, $\mathcal{L}_\mathcal{E}$ can be one of the elements in the set V.

Random Assignment

- (R1) $\{true\}\ x \xleftarrow{\$} \mathcal{U}\ \{\mathsf{Indis}(\nu x) \wedge \mathsf{Icounter}(x; \{x\})\}$
- (R2) $\{\mathsf{Indis}(\nu t; V)\}\ x \xleftarrow{\$} \mathcal{U}\ \{\mathsf{Indis}(\nu t; V, x)\}$
- (R3) $\{\mathsf{E}(\mathcal{E}; t; V)\}\ x \xleftarrow{\$} \mathcal{U}\ \{\mathsf{E}(\mathcal{E}; t; V, x)\}$

Xor Operator

- (X1) $\{\mathsf{Indis}(\nu y; V, y, z)\}\ x := y \oplus z\ \{\mathsf{Indis}(\nu x; V, x, z)\}$ if $y \neq z$ and $y \notin V$
- (X2) $\{\mathsf{Indis}(\nu t; V)\}\ x := y \oplus z\ \{\mathsf{Indis}(\nu t; V)\}$ if $x \notin V$, even if $t = y$ or $t = z$
- (X3) $\{\mathsf{Indis}(\nu t; V, y, z)\}\ x := y \oplus z\ \{\mathsf{Indis}(\nu t; V, x, y, z)\}$

Due to the commutativity of the Xor operation, the role of y and z can be reversed in all the rules above.

Concatenation

- (C1) $\{\mathsf{Indis}(\nu y; V, y, z) \wedge \mathsf{Indis}(\nu z; V, y, z)\}\ x := y\|z\ \{\mathsf{Indis}(\nu x; V, x)\}$ if $y, z \notin V$
- (C2) $\{\mathsf{Indis}(\nu t; V, y, z)\}\ x := y\|z\ \{\mathsf{Indis}(\nu t; V, x, y, z)\}$
- (C3) $\{\mathsf{Indis}(\nu t; V)\}\ x := y\|z\ \{\mathsf{Indis}(\nu t; V)\}$ if $x \notin V$, even if $t = y$ or $t = z$

Increment

- (I1) $\{\mathsf{Icounter}(y; V)\}\ x := y + 1\ \{\mathsf{Icounter}(x; V, x) \wedge \mathsf{E}(\mathcal{E}; x; \mathsf{Var} - x)\}$
- (I2) $\{\mathsf{Indis}(\nu y; V)\}\ x := y + 1\ \{\mathsf{Indis}(\nu x; V)\}$ if $y \notin V$
- (I3) $\{\mathsf{Indis}(\nu t; V)\}\ x := y + 1\ \{\mathsf{Indis}(\nu t; V)\}$ if $x \notin V$ even if $t = y$
- (I4) $\{\mathsf{Indis}(\nu t; V, y)\}\ x := y + 1\ \{\mathsf{Indis}(\nu t; V, x, y)\}$ if $x \notin V$
- (I5) $\{\mathsf{Icounter}(y; V_1) \wedge \mathsf{E}(\mathcal{E}; t; V_2)\}\ x := y + 1\ \{\mathsf{E}(\mathcal{E}; t; V_2, x)\}$ even if $t = y$

Block Cipher

- (B1) $\{\mathsf{E}(\mathcal{E}; y; \emptyset)\}\ x := \mathcal{E}(y)\ \{\mathsf{Indis}(\nu x) \wedge \mathsf{Icounter}(x; \{x\})\}$
- (B2) $\{\mathsf{Indis}(\nu t; V) \wedge \mathsf{E}(\mathcal{E}; y; \emptyset)\}\ x := \mathcal{E}(y)\ \{\mathsf{Indis}(\nu t; V, x)\}$ provided $\mathcal{L}_\mathcal{E} \notin V$ even if $t = y$

– (B3) $\{\mathsf{Indis}(\nu t; V, \mathcal{L}_{\mathcal{E}}, y) \wedge \mathsf{E}(\mathcal{E}; y; \emptyset)\}$ $x := \mathcal{E}(y)$ $\{\mathsf{Indis}(\nu t; V, \mathcal{L}_{\mathcal{E}}, x, y)\}$
– (B4) $\{\mathsf{Icounter}(t; V) \wedge \mathsf{E}(\mathcal{E}; y; \emptyset)\}$ $x := \mathcal{E}(y)$ $\{\mathsf{Icounter}(t; V)\}$ even if $t = y$
– (B5) $\{\mathsf{E}(\mathcal{E}; t; V, y)\}$ $x := \mathcal{E}(y)$ $\{\mathsf{E}(\mathcal{E}; t; V, y)\}$

Finally, we add a few rules whose purpose is to preserve invariants that are unaffected by the command.

Generic Preservation Rules
Assume that $t \neq x, y, z$ and c is either $x \xleftarrow{\$} \mathcal{U}$, $x := y\|z$, $x := y \oplus z$, or $x := w+1$:

– (G1) $\{\mathsf{Icounter}(t; V)\}$ c $\{\mathsf{Icounter(t;V)}\}$ if $y, z \notin V$
– (G2) $\{\mathsf{E}(\mathcal{E}; t; V)\}$ c $\{\mathsf{E}(\mathcal{E}; t; V)\}$ if $x \notin V$, even if $t = y$ or $t = z$

4 Example

We show in Figure 2 how our new rules can be used to prove the security of the PCBC encryption mode. We only consider messages consisting of 3 blocks, as it would be clear that the propagaion of invariants would continue for any finite number of blocks.

To simplify the exposition, we only show the invariants that are necessary for the proof of security. An automated verifier would obtain the proof of security by applying every applicable rule at each execution step. In all the examples, we denote by, say, (L3) the use of the third part of Lemma 1.

$\mathcal{E}_{PCBC}(m_1|m_2|m_3, IV|c_1|c_2|c_3)$
var $IV, z_1, z_2, z_3, y_2, y_3;$

$IV \xleftarrow{\$} \mathcal{U};$	$\{\mathsf{Indis}(\nu IV)\}$	(R1)
$z_1 := IV \oplus m_1;$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var}^* - z_1) \wedge \mathsf{Indis}(\nu z_1; \mathsf{Var}^* - IV)$	(X1)(X2)
	$\wedge\ \mathsf{E}(\mathcal{E}; z_1; \mathsf{Var} - IV - z_1)\}$	(L1)
$c_1 := \mathcal{E}(z_1);$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1)\}$	(B1)(B2)
$y_2 := c_1 \oplus m_1;$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1; \mathsf{Var}^* - y_2)$	(X2)(X3)
	$\wedge\ \mathsf{Indis}(\nu y_2; \mathsf{Var}^* - c_1)\}$	(X1)
$z_2 := y_2 \oplus m_2;$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1; \mathsf{Var}^* - y_2 - z_2)$	(X2)(X3)
	$\wedge\ \mathsf{Indis}(\nu z_2; \mathsf{Var}^* - c_1 - y_2) \wedge \mathsf{E}(\mathcal{E}; z_2; \mathsf{Var} - c_1 - y_2 - z_2)\}$	(X1)(L1)
$c_2 := \mathcal{E}(z_2);$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1; \mathsf{Var} - y_2 - z_2)$	(B2)
	$\wedge\ \mathsf{Indis}(\nu c_2)\}$	(B1)
$y_3 := c_2 \oplus m_2;$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1; \mathsf{Var} - y_2 - z_2)$	(X3)
	$\wedge\ \mathsf{Indis}(\nu c_2; \mathsf{Var}^* - y_3) \wedge \mathsf{Indis}(\nu y_3; \mathsf{Var}^* - c_2)\}$	(X1)(X2)
$z_3 := y_3 \oplus m_3;$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1; \mathsf{Var} - y_2 - z_2)$	(X3)
	$\wedge\ \mathsf{Indis}(\nu c_2; \mathsf{Var}^* - y_3 - z_3) \wedge \mathsf{Indis}(\nu z_3; \mathsf{Var}^* - c_2 - y_3)$	(X1)(X2)
	$\wedge\ \mathsf{E}(\mathcal{E}; z_3; \mathsf{Var} - c_2 - y_3 - z_3)$	(L1)
$c_3 := \mathcal{E}(z_3);$	$\{\mathsf{Indis}(\nu IV; \mathsf{Var} - z_1) \wedge \mathsf{Indis}(\nu c_1; \mathsf{Var} - y_2 - z_2)$	(B2)
	$\wedge\ \mathsf{Indis}(\nu c_2; \mathsf{Var}^* - y_3 - z_3) \wedge \mathsf{Indis}(\nu c_3)\}$	(B1)(B2)

Fig. 2. Analysis of PCBC encryption mode

5 Conclusion

We improved on the result of Gagné et al. [9] by proposing a new Hoare logic with more precise invariants and more complete rule set. This logic can be used to construct an automated verification tool that can successfully verify the security of all the symmetric encryption modes that could be verified by [9], in addition to many more that it could not.

Future directions to this work include the addition of loops to our grammar to remove the necessity of having a different program for each message length. We would also like to use a similar system to model other security properties, such as unforgeability (for message authentication codes) and collision-resistance (for hash functions). We believe that the study of message authentication codes would be of particular interest since, combined with semantically secure encryption, it would allow us to prove the chosen-ciphertext (CCA) security of certain symmetric authenticated encryption modes.

References

1. Barthe, G., Daubignard, M., Kapron, B., Lakhnech, Y.: Computational indistinguishability logic. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, pp. 375–386. ACM (2010)
2. Barthe, G., Grégoire, B., Lakhnech, Y., Béguelin, S.Z.: Beyond Provable Security Verifiable IND-CCA Security of OAEP. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 180–196. Springer, Heidelberg (2011)
3. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Annual IEEE Symposium on Foundations of Computer Science, p. 394 (1997)
4. Chakraborty, D., Nandi, M.: An improved security bound for HCTR, pp. 289–302 (2008)
5. Chakraborty, D., Sarkar, P.: A New Mode of Encryption Providing a Tweakable Strong Pseudo-Random Permutation. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 293–309. Springer, Heidelberg (2006)
6. Chakraborty, D., Sarkar, P.: HCH: A new tweakable enciphering scheme using the hash-counter-hash approach. IEEE Transactions on Information Theory 54(4), 1683–1699 (2008)
7. Courant, J., Daubignard, M., Ene, C., Lafourcade, P., Lahknech, Y.: Towards automated proofs for asymmetric encryption schemes in the random oracle model. In: Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008), Alexandria, USA (October 2008)
8. Gagné, M., Lafourcade, P., Lakhnech, Y., Safavi-Naini, R.: Automated verification of block cipher modes of operation, an improved method. Technical Report TR-2011-9, Laboratoire Verimag, Université Joseph Fourier, France, 21 pages (April 2011), `http://www-verimag.imag.fr/~gagne/TechRep2011_09.pdf`
9. Gagné, M., Lafourcade, P., Lakhnech, Y., Safavi-Naini, R.: Automated Security Proof for Symmetric Encryption Modes. In: Datta, A. (ed.) ASIAN 2009. LNCS, vol. 5913, pp. 39–53. Springer, Heidelberg (2009)
10. Halevi, S.: EME*: Extending EME to Handle Arbitrary-Length Messages with Associated Data. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 315–327. Springer, Heidelberg (2004)

11. Halevi, S.: Invertible Universal Hashing and the Tet Encryption Mode. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 412–429. Springer, Heidelberg (2007)
12. Halevi, S., Rogaway, P.: A Tweakable Enciphering Mode. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer, Heidelberg (2003)
13. Halevi, S., Rogaway, P.: A Parallelizable Enciphering Mode. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer, Heidelberg (2004)
14. Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129–153. Springer, Heidelberg (2003)
15. Iwata, T., Kurosawa, K.: On the Security of a New Variant of OMAC. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 67–78. Springer, Heidelberg (2004)
16. Iwata, T., Kurosawa, K.: Stronger Security Bounds for OMAC, TMAC, and XCBC. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 402–415. Springer, Heidelberg (2003)
17. Jutla, C.S.: Encryption Modes with Almost Free Message Integrity. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 529–544. Springer, Heidelberg (2001)
18. Kurosawa, K., Iwata, T.: TMAC: Two-key CBC MAC. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 33–49. Springer, Heidelberg (2003)
19. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable Block Ciphers. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer, Heidelberg (2002)
20. McGrew, D.A., Fluhrer, S.R.: The security of the extended codebook (XCB) mode of operation (2007)
21. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (2004)
22. Wang, P., Feng, D., Wu, W.: On the Security of Tweakable Modes of Operation: TBC and TAE. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 274–287. Springer, Heidelberg (2005)