Completeness Theorems for the Abadi-Rogaway Language of Encrypted Expressions

Daniele Micciancio   Bogdan Warinschi*

Computer Science Department,
University of California, San Diego
9500 Gilman Drive, CA 92093
{daniele,bogdan}@cs.ucsd.edu

Abstract

We show that the Abadi-Rogaway logic of indistinguishability for cryptographic expressions is not complete, giving a natural example of encryption function (satisfying the standard notion of security) and a pair of expressions, such that the distributions associated to the two expressions are computationally indistinguishable, but equality cannot be proved within the Abadi-Rogaway logic. Then, we show that if a stronger, yet natural, notion of security for encryption is used (namely, that of authenticated encryption), then the Abadi-Rogaway logic is both sound and complete. In addition, we consider a refinement of the Abadi-Rogaway logic that overcomes certain limitations of the original proposal, allowing for encryption functions that do not hide the length of the message being sent. Both the soundness theorem of Abadi and Rogaway, and our completeness result for authenticated encryption easily extend to this more realistic notion of secrecy.

1 Introduction

In the last two decades, the logic and complexity theory communities have developed two related, though quite different, approaches to the theoretical modeling of security. Each approach has its own benefits and shortcomings. The logic approach, usually based on some formal language, is primarily intended to make the task of writing and verifying cryptographic protocols more manageable. However, this approach is often criticized because it is not clear how well the semantics of the language captures the right intuition of cryptographic security.

On the other hand, formalizing and clarifying the meaning of common security notions like *secrecy*, *authenticity*, etc., has been one of the primary goals and major achievements of the complexity based approach to cryptography. The stronger and more comprehensive adversarial model used by this approach has gained wide acceptance in the cryptographic community as the one providing the “right” notion of security. However, the relatively unstructured way in which security is formalized in this framework, makes proving the security of even moderately complex protocols quite a formidable task.

Recently, several attempts have been made to bridge the gap between these two worlds [10, 7, 1, 9]. Still, most of these approaches do not retain the simplicity of the original logic formulations, or do not capture the complexity theoretic notion of security in its full power. In a recent paper [2], Abadi and Rogaway defined a language of cryptographic expressions, and gave a simple semantics such that if two expressions have the same semantics then they are computational indistinguishable. The language **Exp** considered in [2] consists of all expressions that can be built from basic messages and cryptographic keys, using concatenation and

*Authors were supported in part by NSF Career Award CCR-0093029

encryption operation, according to the grammar:

$$\begin{array}{rlll} M ::= & K & \text{key (for K} \in \textbf{Keys}) \\ & i & \text{bit (for i} \in \textbf{Bool}) \\ & (M, M) & \text{pair} \\ & \{M\}_K & \text{encryption(for K} \in \textbf{Keys}) \end{array} \tag{1}$$

Abadi and Rogaway also give a simple (and easily computable) semantics that associates to each expression E an element of the language **Pattern** which extends the expression grammar with a special symbol used to denote undeciphreable messages:

$$\begin{array}{rlll} P ::= & K & \text{key (for K} \in \textbf{Keys}) \\ & i & \text{bit (for i} \in \textbf{Bool}) \\ & (P, Q) & \text{pair} \\ & \{P\}_K & \text{encryption(for K} \in \textbf{Keys}) \\ & \Box & \text{undecryptable} \end{array} \tag{2}$$

(see [2] for a definition of the pattern evaluation function). The main result of [2] is that if two expressions have the same pattern, then the probability distributions naturally associated to the two expressions (when keys are chosen independently at random, and encryption scheme satisfies the standard notion of security) are computationally indistinguishable. The standard notion of security for encryption functions [6] is that for any two messages m_0 and m_1, and randomly chosen key k, the two ciphertexts $E_k(m_0)$ and $E_k(m_1)$ are computationally indistinguishable. The intuition behind this definition is that not only recovering the plaintext m from $E_k(m)$ must be hard, but also computing any partial information about m should be infeasible. Similarly, the indistinguishability of the probability distributions associated to two expressions with the same pattern, captures the computational notion that the encoded expressions do not leak any partial information other than what was intended by the protocol designer, as specified by the pattern.

The cryptographic language (1) is clearly limited as it allows to model only the simplest cryptographic protocols where one party sends a sequence of (complex) messages to another one. However, [2] is a nice example of a simple language with a clean semantics which is sound according to the standard notion of security of complexity based cryptography. So, [2] is an interesting first step toward languages that can be used to model more complex cryptographic protocols. In this paper we give a closer look at the semantics of [2], and consider the following two important issues:

- Abadi and Rogaway consider encryption functions that hide every possible information about the message. This is not a realistic assumption, because in order to allow unique decryption, the ciphertext must be at least as long as the plaintext. In particular, the length of the message being encrypted cannot be completely hidden without making the scheme totally impractical.

- Another important issue, is that of completeness, i.e., if the probability associated to two expressions are indistinguishable for every secure encryption function, is it true that the two expressions have the same pattern? In other words, we ask whether the pattern associated to an expression fully characterizes the information recoverable by an adversary, or it is possible for two expressions to be equivalent in the computational setting, and still equivalence cannot be proved using the Abadi-Rogaway pattern semantics.

Regarding the first issue, as clearly stated in [2], this is only a simplifying assumption to make the exposition more readable, and their semantics and soundness theorem can be extended to more realistic encryption schemes that hide all information about the plaintext other than its length.

Regarding the second point, [2] already gives an example to illustrate that their semantics is not complete. However, the example is quite artificial because it involves an encryption function with restricted message space. If we consider general purpose encryption schemes (i.e., schemes that can be used to encrypt arbitrary messages), then the example of [2] breaks down, leaving the question of completeness open. In this paper we show that the Abadi-Rogaway pattern semantics is not complete, even for general purpose encryption schemes satisfying the standard notion of security. The example is presented in Section 2. Then, in Section 3, we consider a stronger notions of security for encryption schemes. Namely, we consider *authenticated encryption schemes*, i.e., encryption schemes providing both secrecy and authenticity. This is a stronger, yet natural and relatively standard, notion of security and it has been the subject of many recent research papers in the cryptographic community [5, 3, 8]. Interestingly, we can show that if authenticated encryption is used, then the Abadi-Rogaway pattern semantics is both sound and complete. This is the main technical contribution of this paper, and it is proved in Section 3.

Finally, in Section 4, we extend the pattern semantics of [2] to deal with encryption schemes that do not hide the length of the message, and observe that both the soundness and completeness results hold for these more realistic encryption schemes as well.

2 Incompleteness of the Abadi-Rogaway semantics

Abadi and Rogaway define a function $rec(\cdot)$ that associates to every expression E the set of keys recoverable from E (see [2] for a formal definition). For example

$rec((\{K_1\}_{K_3}, \{0\}_{K_2}, K_1, \{K_2\}_{K_1})) = \{K_1, K_2\}$

The function $patternep : \mathbf{Exp} \times \mathcal{P}(\mathbf{Keys}) \to \mathbf{Pattern}$, also introduced in [2], associates to every pair $(E, T) \in \mathbf{Exp} \times \mathcal{P}(\mathbf{Keys})$ the pattern obtained by replacing encryptions under keys outside the set T with the $\Box$ symbol. If E is as in the previous example,

$$patternep(E, \{K_2\}) = (\Box, \{0\}_{K_2}, K_1, \Box)$$

Such a pattern models the partial information an adversary can get from E having the knowledge of the keys in T. Finally, the pattern associated to an expression is defined as $patternep(E, rec(E))$ and two cryptographic expressions E_1, E_2 are deemed equivalent if their associated patterns are equal up to renaming of the keys. We write $E_1 \cong E_2$ to denote this equivalence.

From a computational perspective, a symmetric encryption scheme Π is given by three algorithms: *the key generation algorithm* $\mathcal{K}$, takes as input a security parameter η and outputs a key k in some domain Key. The *encryption algorithm* $\mathcal{E}$ takes as input a key k and a *plaintext* m and outputs a *ciphertext* $\mathcal{E}_k(m)$. Finally, the *decryption algorithm* $\mathcal{D}$, takes as input a key k and a string c and outputs the underlying plaintext or $\perp$ (used to indicate decryption failures.) It is required that for any string m and any key k, $\mathcal{D}_k(\mathcal{E}_k(m)) = m$.

For an encryption scheme Π, fixing a security parameter η induces a natural mapping $[\![\cdot]\!]_{\Pi(\eta)}$ from expressions to probability distributions on strings. For an expression E, random keys are assigned to the key names appearing in E. Then, the distribution is inductively defined following the structure of E.

Assuming what is called in [2] a type-0 secure encryption scheme Π (namely an encryption scheme that hides all partial information on the key, and all partial information on the plaintext) the following connection can be shown between the formal and the computational approach (acyclic expressions are those with no encryption cycles):

Theorem 2.1 Let E_0 and E_1 be acyclic expressions and let Π be a type-0 secure encryption scheme. Suppose that $E_0 \cong E_1$, then $[\![E_0]\!]_{\Pi(\eta)} \approx [\![E_1]\!]_{\Pi(\eta)}$.

Next we provide a counterexample for the converse of the theorem which sheds some light on the properties that might be needed in order to obtain the converse result. Since our example is based on a real encryption scheme, the encryption function does not hide the length of the message. However, this is not important because all encrypted messages in the example have the same length.

Consider the following encryption scheme based on a family $\{F_i\}_{i \in I}$ of pseudorandom functions. The key generation algorithm simply outputs a random index $k \in I$. The encryption of a message m is $\mathcal{E}_k(m) = r || F_k(r) \oplus m$, where r is some generated randomness, and $||$ is string concatenation. The decryption of a message $(r||c)$ is simply $F_k(r) \oplus c$. Consider the expressions

$E_1 = (\{K_3\}_{K_1}, K_1)$ and $E_2 = (\{K_3\}_{K_1}, K_2)$.

Then, while the patterns associated:

$pattern(E_1) = (\{K_3\}_{K_1}, K_1)$

$pattern(E_2)) = (\Box, K_2)$.

are clearly different, the distributions of $(r_1||F_{k_1}(r_1) \oplus k_3, k_1)$ and $(r_2||F_{k_1}(r_2) \oplus k_3, k_2)$ are computationally indistinguishable. In fact, they are identical when r_1, r_2, k_1, k_2, k_3 are picked at random. This is related to the fact that knowledge of a key k does not imply the ability of distinguishing valid decryptions under k from invalid ones.

3 A completeness theorem for authenticated encryption

Next we consider secure authenticated encryption schemes. Informally, such schemes provide a mechanism for checking the validity of a ciphertext. The authenticity of the plaintext is ensured, if no efficient adversary can create a valid ciphertext except with negligible probability. The formal definition below, follows [3, 5].

Definition 3.1 [Integrity of an Authenticated Encryption Scheme] Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme and $\eta \in \mathsf{N}$ a security parameter. Consider the following experiment involving an algorithm A with access to an encryption oracle:

$\mathbf{Exp}^{\text{int-ptxt}}_{\Pi,A}(\eta)$
  $K \xleftarrow{R} \mathcal{K}(\eta)$
  $C \xleftarrow{R} A^{\mathcal{E}_K(.)}(\eta)$
  If $\mathcal{D}_K(C) \neq \perp$ and C was never
  a response of $\mathcal{E}_K(.)$ then return 0
    else return 1

Define the advantage of the algorithm A as: $\mathbf{Adv}^{\text{int-ptxt}}_{\Pi,A}(\eta) = \Pr\left[\mathbf{Exp}^{\text{int-ptxt}}_{\Pi,A}(\eta) = 1\right]$. The encryption scheme Π is said to be INT-PTXT *secure* ("integrity of plaintext secure") if the advantage function $\mathbf{Adv}^{\text{int-ptxt}}_{\Pi,A}(\cdot)$ is negligible for any adversary A whose time-complexity is polynomial in η.

As a consequence, for a INT-PTXT secure encryption scheme, decrypting with key k_1 the encryption of some message under the key k_2 fails with high probability. This is captured by the following lemma:

Lemma 3.2 Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a INT-PTXT secure symmetric authenticated encryption scheme. Then for any $x \in \mathsf{String}$ and $\eta \in \mathsf{N}$, $\Pr_{k_1,k_2 \xleftarrow{R} \mathcal{K}(\eta)}[\,\mathcal{D}_{k_1}(\mathcal{E}_{k_2}(x))\,] \leq \nu_0(\eta)$, for some negligible function $\nu_0(\cdot)$.

Since the proof of the lemma exhibits an adversary which does not make use of the encryption oracle, one could consider weaker security notions for Π. We remark that although most of our statements use encryption schemes that are INT-PTXT secure, we only need schmes satisfying the property of INT-PTXT proved in Lemma 3.2.

3.1 The Completeness Theorem

We show that for authenticated encryption, the pattern semantics agrees with the computational semantics. More precisely we prove the following theorem:

Theorem 3.3 Let Π be an INT-PTXT secure encryption scheme. Then if E_0 and E_1 are acyclic expressions such that $E_0 \ncong E_1$ then $[\![E_0]\!]_{\Pi(\eta)} \not\approx [\![E_1]\!]_{\Pi(\eta)}$.

The proof is based on the intuition that from a sample from $[\![E]\!]_{\Pi(\eta)}$ one can recover the pattern of E. Next we define some auxiliary functions that will prove to be useful in reasoning about the recovery algorithm.

The function $F_{kr} : \mathbf{Exp} \times \mathcal{P}(\mathbf{Keys}) \to \mathcal{P}(\mathbf{Keys})$ defined by

- $F_{kr}(i, T) = T$;
- $F_{kr}(K, T) = \{K\} \cup T$;
- $F_{kr}((E_0, E_1), T) = F_{kr}(E_0, T) \cup F_{kr}(E_1, T)$;
- $F_{kr}(\{E\}_K, T) = T$, if $K \notin T$;
- $F_{kr}(\{E\}_K, T) = F_{kr}(E, T)$, if $K \in T$;

computes the set $F_{kr}(E,T)$ containing the keys that can be recovered from E having knowledge of the keys in T going through E only once. Using the auxiliary functions $G_i : \mathbf{Exp} \to \mathcal{P}(\mathbf{Keys})$ defined by the relations:

- $G_0(E) = \emptyset$;
- $G_i(E) = F_{kr}(E, G_{i-1}(E))$

we define F_{ikr}, the iterated version of F_{kr} as:

$$F_{ikr}(E) = \bigcup_i G_i(E).$$

The following equality holds: $rec(E) = F_{ikr}(E)$.

Example. If $E = (\{\{K_2\}_{K_1}\}_{K_1}, \{K_3\}_{K_2}, K_1)$ then $F_{kr}(E,\emptyset) = \{K_1\}$ and $F_{kr}(E,\{K_1\}) = \{K_1, K_2\}$. According to the definition of G_i, $G_1(E) = \{K_1\}$ and $G_2(E) = \{K_1, K_2\}$. Also, $F_{ikr}(E) = \{K_1, K_2\}$.

The intuition that when given a sample from the distribution associated to an expression E one can recover partial information (e.g. keys that are sent in clear), is captured by the following two functions. The key recovery function $\mathsf{kr} : \mathsf{String} \times \mathcal{P}(\mathsf{Key}) \to \mathcal{P}(\mathsf{Key})$ and its iterated version $\mathsf{ikr} : \mathsf{String} \to \mathsf{Key}$ are defined by the following two algorithms:

Algorithm $\mathsf{kr}(e,T)$
    if $e = \langle i, \text{“}bool''\rangle$ output T
    if $e = \langle k, \text{“}key''\rangle$ output $T \cup \{k\}$
    if $e = \langle (m,n), \text{“}pair''\rangle$ output $\mathsf{kr}(m,T) \cup \mathsf{kr}(n,T)$
    if $e = \langle c, \text{“}ciphertext''\rangle$
        if for exactly one $k \in T, \mathcal{D}_k(y) \neq \perp$
        then output $\mathsf{kr}(\mathcal{D}_{k_i}(c))$
        else output T

Algorithm $\mathsf{ikr}(e)$
    $T_0 \leftarrow \emptyset$; $cont \leftarrow true$
    while $cont$
        $T_{i+1} \leftarrow \mathsf{kr}(e, T_i)$
        if $T_{i+1} = T_i$ then $cont \leftarrow false$
            else $i \leftarrow i + 1$
    output T_i

The relation between these two functions and their syntactic counterparts (F_{kr} and F_{ikr}) is established by the following two lemmas (their proof as well as the proofs of other lemmas are omitted in this abstract due to space constraints). The intuition is simple: we show that the functions $\mathsf{kr}()$ and $\mathsf{ikr}()$ are such that the following two diagrams commute:

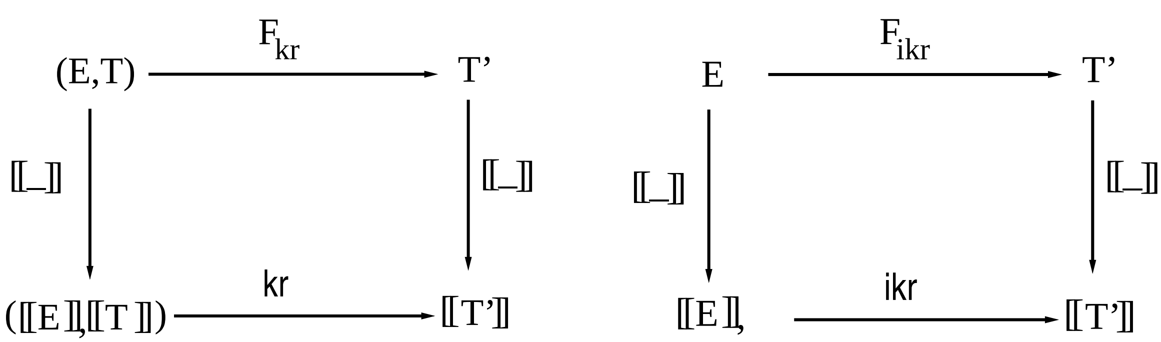

Figure 1: Relation between syntactic and computational

Lemma 3.4 For any INT-PTXT secure authenticated encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, any expression $E \in \mathbf{Exp}$, any set of keys $T \subseteq \mathbf{Keys}$ and any security parameter $\eta \in \mathsf{N}$ let $\nu_0(\cdot)$ be as in Lemma 3.2. Then: $\Pr[\, \mathsf{kr}(e, \tau(T)) \neq \{\tau(F_{kr}(E, T)\})\,] \leq |E||T|\nu_0(\eta)$. where the probability is taken over $e \xleftarrow{R} [\![E]\!]_{\Pi(\eta)}$.

Lemma 3.5 For any INT-PTXT secure authenticated encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, any $E \in \mathbf{Exp}$ and any security parameter $\eta \in \mathsf{N}$, $\Pr_{e \xleftarrow{R} [\![E]\!]_{\Pi(\eta)}}[\, \mathsf{ikr}(e) \neq \tau(F_{ikr}(E))\,] \leq |E|^3 \nu_0(\eta)$

In a similar manner we associate a computational counterpart to the function *patternep* described in Section 2. Let $\mathsf{psp} : \mathsf{String} \times \mathcal{P}(\mathsf{Key}) \to \mathbf{Pattern}$ be the function computed by the following algorithm:

Algorithm $\mathsf{psp}(e, tT)$
  let $f : tT \to \mathbf{Keys}$ be an injective function
  if $e = \langle i, \text{“}bool''\rangle$ output i;
  if $e = \langle k, \text{“}key''\rangle$ and $k \in tT$ then output $f(k)$;
  if $e = \langle (m, n), \text{“}pair''\rangle$ output $(\mathsf{psp}(m, tT), \mathsf{psp}(n, tT))$;
  if $e = \langle c, \text{“}ciphertext''\rangle$
    if for exactly one key $k \in tT, \mathcal{D}_k(c) \neq \perp$
    output $\{\mathsf{psp}(\mathcal{D}_k(c), tT)\}_{f(k)}$
    else output $\Box$;

The following connection between *patternep* and psp can be shown:

Lemma 3.6 For any secure authenticated encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, for any expression $E \in \mathbf{Exp}$, any set of key names $T \subseteq \mathbf{Keys}$ and any security parameter η, $\Pr[\, \mathsf{psp}(e, \tau(T)) \ncong patternep(E, T)\,] \leq |E||T|\nu_0(\eta)$, where the probability is taken on $\epsilon \xleftarrow{R} [\![E]\!]_{\Pi(\eta)}$ and $\nu_0(\cdot)$ is as in Lemma 3.2.

We can give now the proof of Theorem 3.3.

Proof:

Consider two arbitrary expressions $E_0, E_1 \in \mathbf{Exp}$ such that $E_0 \ncong E_1$. We will exhibit an algorithm that distinguishes between $[\![E_0]\!]_{\Pi(\eta)}$ and $[\![E_1]\!]_{\Pi(\eta)}$ with non-negligible probability:

Let A be the following algorithm:

Algorithm $A(e)$
  $tT \leftarrow \mathsf{ikr}(e)$
  if $\mathsf{psp}(e, tT) \cong p(E_0, rec(E_0))$ then output 0
    else output 1

For $e_0 \xleftarrow{R} [\![E_0]\!]_{\Pi(\eta)}$, denote by A_i the event that $\mathsf{psp}(e_i, tT) \cong p(E_i, rec(E_i))$, for $i = 1, 2$. Then:

$\Pr[\, A_0\,] \geq \Pr[\, A_0 \wedge (tT = \tau(rec(E_0)))\,] =$

$\Pr[\, A_0 \mid tT = \tau(rec(E_0))\,] \Pr[\, \mathsf{ikr}(e) = \tau(rec(E_0))\,]$

From Lemma 3.6 and Lemma 3.5 it follows that $\Pr[\, A_0 \mid tT = \tau(rec(E_0))\,] \geq (1 - |E_0||tT|\nu_0(\eta))$ and $\Pr[\, \mathsf{ikr}(e) = \tau(rec(E_0))\,] \geq (1 - |E|^3\nu_0(\eta))$. In a similar manner we can show that $\Pr[\, A_1\,] \geq (1 - 2|E_1|^3\nu_0(\eta)$. Then:

$$\Pr[\, A(e_0) = 0\,] = \Pr[\, A_0\,] \geq 1 - 2|E_0|^3\nu_0(\eta). \tag{3}$$

$E_0 \ncong E_1$ implies $p(E_0, rec(E_0)) \ncong p(E_1, rec(E_1))$. Therefore, when $e_1 \xleftarrow{R} [\![E_1]\!]_{\Pi(\eta)}$,

$\Pr[\, A(e_1) = 0\,] =$

$\Pr[\, \mathsf{psp}(e_1, tT) \cong p(E_0, rec(E_0))\,] =$

$(1 - \Pr[\, \mathsf{psp}(e_1, tT) \ncong p(E_0, rec(E_0))\,]) \leq$

$1 - \Pr[\, \mathsf{psp}(e_1, tT) \cong p(E_1, rec(E_1))\,] = 1 - \Pr[\, A_1\,]$ and we obtain:

$$\Pr[\, A(e_1) = 0\,] \leq 2|E_1|^3\nu_0(\eta) \tag{4}$$

From (3) and (4) if follows that A distinguishes between $[\![E_0]\!]_{\Pi(\eta)}$ and $[\![E_1]\!]_{\Pi(\eta)}$ with probability greater than:

$$1 - 2(|E_0|^3 + |E_1|^3)\nu_0(\eta)$$

which is non-negligible. ▮

4 Encryption schemes that leak the length of the message

In this section we show the validity of Theorems 2.1 and Theorem 3.3 for weaker but more reasonable security assumptions. More precisely we show that for encryption schemes which do not hide the length of the plaintext, a simple modification of the pattern semantics preserves the above mentioned results.

We make the assumptions that basic messages are encoded in blocks of length equal to the length of the key, and also that a ciphertext is one block longer than the underlying plaintext. It is important to note that these assumptions are not significant simplifications, since most of the symmetric encryptions schemes that are currently used in practice (CBC, CTR [4]), have these property. Also, it is easy to see how to extend our treatment to a more general setting when some relation between the lengths of the plaintext and ciphertext is known.

While the language of expressions remains the same, we modify the language of patterns by replacing $\Box$ with natural numbers. The intuition is that the knowledge an adversary has may contain information of the type: "undecryptable ciphertext of length n blocks" rather than simply "undecryptable ciphertext". The grammar for patterns becomes:

$$\begin{array}{lll} P ::= & K & \text{key (for K} \in \mathbf{Keys}) \\ & i & \text{bit (for i} \in \mathbf{Bool}) \\ & (M, N) & \text{pair} \\ & \{M\}_K & \text{encryption(for K} \in \mathbf{Keys}) \\ & n & \text{undecryptable ciphertext} \\ & & \text{of length n} \end{array} \tag{5}$$

Next, consider the function $L : \mathbf{Exp} \to \mathsf{N}$ defined inductively as follows:

- $L(K) = 1$ for any $K \in \mathbf{Keys}$;
- $L(B) = 1$ for any $B \in \mathbf{Blocks}$;
- $L((E_0, E_1)) = L(E_0) + L(E_1)$ for any $E_0, E_1 \in \mathbf{Exp}$;
- $L(\{E\}_K) = L(E) + 1$;

We change the definition of *patternep* by replacing *patternep*$(\{E\}_K, T) = \Box$ with *patternep*$(\{E\}_K, T) = L(E)$ for the case when $K \notin T$. Expression equivalence remains the same: two expressions are equivalent if their associated patterns are equal up to renaming. The computational semantics remains essentially the same as in [2].

It is not hard to see that while the statement and the proof of Theorem 3.3 remain essentially the same, the hypothesis of Theorem 2.1 can now be weaken to include "type-1" secure encryption schemes (encryption schemes hiding all partial information about the keys and all partial information about encrypted plaintexts except possibly their length):

Theorem 4.1 Let E_0 and E_1 be acyclic expressions and let Π be a type-1 secure encryption scheme. Suppose that $E_0 \cong E_1$, then $[\![E_0]\!]_{\Pi(\eta)} \approx [\![E_1]\!]_{\Pi(\eta)}$.

References

[1] M. Abadi and J.Jűrgens. Formal eavesdropping and its computational interpretation. In *Proceedings of the Fourth International Symposium on Theoretical Aspects of Computer Software (to appear).*

[2] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP TCS*, pages 3–22, 2000.

[3] J. An and M. Bellare. Does encryption with redundancy provide authenticity? In *Advances in cryptology — EUROCRYPT 2001*, number 2045 in Lecture Notes in Computer Science, pages 512–528, Berlin, Germany, 2001. Springer-Verlag.

[4] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in cryptology - CRYPTO 1998*, number 1462 in Lecture Notes in Computer Science, pages 26–45. Springer-Verlag, 1998.

[5] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic c omposition paradigm. In *Advances in cryptology-ASIACRYPT*, number 1976 in Lecture Notes in Computer Science, pages 531–545. Springer-Verlag, 2000.

[6] S. Goldwasser and S. Micali. Probabilisti encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[7] J. Guttman, F. J. Thayer, and L. Zuck. The faithfulness of abstract encryption. In *Proceedings of The 8th ACM Conference on Computer and Communication Security, 2001 (to appear).*

[8] H. Krawczyk. The order of encryption and authentication for protecting communication. In *Advances in cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, Berlin, Germany, 2001. Springer-Verlag.

[9] P. Lincoln, J. Mitchell, M. Mitchell, and A. Scedrov. A probabilistic poly-time framework for protocol analysis. In *Proceedings of the Fifth ACM Conference on Computer and Communications Security*, pages 112–121, 1998.

[10] B. Pfitzmann, M. Schunter, and M. Waidner. Cryptographic security of reactive systems. In *Electronic Notes in Theoretical Computer Science, 32, April 2000.*