Algorithmic Problems in the Symbolic Approach to the Verification of Automatically Synthesized Cryptosystems

Hai Lin[1], Christopher Lynch[1], Andrew M. Marshall[2(✉)], Catherine A. Meadows[3], Paliath Narendran[4], Veena Ravishankar[2(✉)], and Brandon Rozek[2]

[1] Clarkson University, Potsdam, NY, USA
[2] University of Mary Washington, Fredericksburg, VA, USA
{amarsha2,vravisha}@umw.edu
[3] Naval Research Laboratory, Washington, DC, USA
[4] University at Albany–SUNY, Albany, NY, USA

Abstract. Automated methods can be used to generate cryptosystems by combining the primitives in an arbitrary fashion, to weed out insecure cryptosystems, and to prove the security of those that survive. In this paper, we study several algorithmic problems arising from the verification of automatically synthesized cryptosystems built from block ciphers, in a theory that includes *ACUN*. One of these is static equivalence to an algorithm that produces a sequence of random terms. The other is invertibility, the problem of determining whether, given an automatically synthesized cryptosystem, built from block ciphers, and the ability to compute inverses, is it always possible to compute the original plaintext from the ciphertext? We show that static equivalence to random in this theory is undecidable in general. In addition, we identify a reasonable special case for which there is a decidable condition implying security, along with an algorithm for verifying it. For invertibility, we identify a reasonable class of cryptosystems for which invertibility is equivalent to a simple syntactic condition that can be easily verified.

Keywords: Cryptographic modes of operation · Symbolic reasoning · Equational theories · Unification

1 Introduction

In this paper we address symbolic analysis problems that arise from the automatic generation and verification of cryptosystems. In this approach one starts with a class of cryptosystems that use a fixed set of functions to combine a fixed set of primitives. Automated methods can be used to generate cryptosystems by

This work was funded by ONR Code 311. The work of Lin, Lynch, Marshall, Narendran, Ravishankar, and Rozek, was funded via NRL grant number N00173-19-1-G012.

B. Konev and G. Reger (Eds.): FroCoS 2021, LNAI 12941, pp. 253–270, 2021.
https://doi.org/10.1007/978-3-030-86205-3_14

combining the primitives in an arbitrary fashion, to weeding out insecure cryptosystems, and proving the security of those that survive. Symbolic techniques have proved particularly helpful in this process, because they give a compact representation of cryptosystems that is amenable to automated analysis.

In this paper we apply a technique we are developing for the synthesis and analysis of *cryptographic modes of operation*. Basic encryption algorithms such as AES are generally *block ciphers* that map λ-bit blocks to λ-bit blocks. A mode of operation combines multiple computations of block cipher encryption to encrypt longer messages securely. We model this block cipher approach by defining a protocol modeling the interaction between an adversary and an encryptor. In this model the adversary sends plaintext blocks, which the encryptor then processes according to some pre-determined method, e.g., the method of a particular cipher. When there are multiple actions that the encryptor can take, the choice is made by the adversary. The encrypted blocks are then sent back to the adversary based on some schedule, e.g. as soon as possible, or only after all the plaintext has been received. It is shown in [6,12], that both the processing method and the schedule are relevant to the security of the cryptosystem.

We consider two symbolic properties. The first is static equivalence [2], between a protocol in which a plaintext-adaptive adversary interacts with a real encryptor, and one in which it interacts with a random encryptor that sends randomly generated blocks. A plaintext-adaptive adversary is one that uses ciphertext it has received previously from an encryptor to construct new plaintext. Static equivalence between two symbolically defined protocols, roughly speaking, requires that, for any trace of one protocol, there is a trace of the other protocol such that any adversarial-computable equation satisfied by the first trace is satisfied by the other, and vice versa. Static equivalence to random may be thought of as the symbolic analog of IND$-CPA security [12], which requires that the cipher text received by the adversary be indistinguishable from a string of random bits.

The second symbolic property, invertibility, requires that a principal able to compute f (the block encryption function) and its inverse be able to retrieve plaintext from ciphertext.

Given one of the above symbolic properties, we can divide the questions we ask about it into two classes. In the first case, given a description of a class of ciphertexts, one can ask whether or not any member has that property. In the second, given a cryptosystem, one can ask whether all ciphertexts produced by that cryptosystem have that property. In this paper we focus on the second, more general, property.

Both questions about static equivalence to random are known to be undecidable for arbitrary convergent term rewriting systems [1,3]. In [8] Lin and Lynch present an algorithm that can be used to answer the first type of question for the class of cryptosystems discussed in this paper. In this paper we devote ourselves to the second type of question: given a mode, whether or not every possible sequence of ciphertext produced by it satisfies static equivalence to random. In Sect. 5.1 we show that this problem is undecidable for cryptographic modes of operation in general. Then, in Sect. 5.2 we give a class of cryptosystems for which there is a decidable property implying static equivalence to random, and we give an algorithm for deciding that property.

The rest of the paper is organized as follows. Section 2 provides the necessary background material. Section 3 defines $MOO_{\oplus}$-programs, which we use for symbolic specification of modes of encryption using the $\oplus$ (xor) function. In Sect. 4 we identify a simple syntactically checkable condition for a class of recursively defined modes of encryption, which we show is equivalent to every ciphertext produced by the mode being invertible. Section 5 considers the decision problems described above. Finally, Sect. 6 concludes the paper and describes some open problems.

1.1 Implementation

We are currently developing a new tool designed to manipulate and analyze Cryptographic Modes of Operation. The goal of this new tool is broad, to develop not only a usable analysis tool for a broad family of cryptographic algorithms but to also develop the underlying libraries which could be used in further analysis or in other symbolic analysis tools (https://symcollab.github.io/CryptoSolve/). As part of that tool, several of the algorithms developed in this paper have been implemented. More details of each implementation are given below as appropriate.

2 Preliminaries

2.1 Terms and Substitutions

Given a first-order signature Σ, a countable set of variables N bound by ν, and a countable set of variables X (s.t. $X \cap N = \emptyset$), the set of terms constructed in the normal recursive manner from X, N, and Σ, is denoted by $T(\Sigma, N \cup X)$. The set of free variables in a term t is denoted by $fv(t)$ and the set of bound variables in t is denoted by $fn(t)$. A term t is *ground* if $fv(t) = \emptyset$. In this paper, we follow the convention of the applied pi calculus [2] and use variables bound by ν to stand for randomly chosen bitstrings. For any position p in a term t (including the root position ϵ), $t(p)$ denotes the symbol at position p, $t|_p$ denotes the subterm of t at position p, and $t[u]_p$ denotes the term t in which $t|_p$ is replaced by u. The size of a term t is denoted by $|t|$ and defined in the usual [2] way as follows: $|f(t_1, \ldots, t_n)| = 1 + \sum_{i=1}^{n} |t_i|$ if f is a n-ary function symbol with $n \geq 1$, $|c| = 1$ if $c \in N$, and $|x| = 0$ if $x \in X$.

A substitution σ is an endomorphism of $T(\Sigma, N \cup X)$ mapping free variables to terms, with only finitely many variables not mapped to themselves, denoted by $\sigma = \{x_1 \mapsto t_1, \ldots, x_m \mapsto t_m\}$. Application of a substitution σ to a term t is written $t\sigma$. Given two substitutions θ and σ, the composition $\sigma \circ \theta$ is the substitution denoted here by $\theta\sigma$ and defined such that $x(\theta\sigma) = (x\theta)\sigma$ for any $x \in X$. The domain of σ is $Dom(\sigma) = \{x \in X \mid x\sigma \neq x\}$. The range of σ

is $Ran(\sigma) = \{x\sigma \mid x \in Dom(\sigma)\}$. When θ and σ are two substitutions with disjoint domains and only ground terms in their ranges, then $\theta\sigma = \theta \cup \sigma$. Given a substitution σ and a finite set of free variables $V \subseteq X$, the restriction of σ to V is the substitution denoted by $\sigma_{|V}$ such that $x\sigma_{|V} = x\sigma$ for any $x \in V$ and $x\sigma_{|V} = x$ for any $x \in X \backslash V$.

2.2 Equational Theories

Given a set E of Σ-axioms (i.e., pairs of Σ-terms, denoted by $l = r$), the *equational theory* $=_E$ is the congruence closure of E under the law of substitutivity. For any Σ-term t, the equivalence class of t with respect to $=_E$ is denoted by $[t]_E$. Since $\Sigma \cap N = \emptyset$, the Σ-equalities in E do not contain any bound variables in N. A theory E is *trivial* if $x =_E y$, for two distinct variables x and y. In this paper, all the considered theories are assumed non-trivial.

The Xor Equational Theory. In this paper we will primarily be concerned with the equational theory of Xor, $E_{\oplus}$. This theory can be represented as a combination of a rewrite system, $R_{\oplus}$, and an associative and commutative equational theory, AC. $E_{\oplus} = R_{\oplus} \cup AC$: $R_{\oplus} = \{x \oplus x \rightarrow 0,\ x \oplus 0 \rightarrow x\}$, $AC = AC(\oplus)$, over the signature, $\Sigma_{\oplus} = \{\oplus, f, 0\}$. We will often use $MOO_{\oplus}$-term to denote a term over $\Sigma_{\oplus}$.

A rewrite rule $\ell \rightarrow r$ is applied to a term t by finding a subterm s of t and a match σ of l and s, i.e., a unifier of l and s that leaves s unchanged, and then replacing s with $r\sigma$. We say that a term is in *normal form* if no rewrite rule can be applied. We note that any term in the $E_{\oplus}$ theory is reducible via a finite set of rewrite rules to a normal form term that is unique up to AC equivalence. If S is finite and $S \subset T_{E_{\oplus}}(\Sigma_{\oplus}, N \cup X)$, and $t \in T_{E_{\oplus}}(\Sigma_{\oplus}, N \cup X)$ we say that $S \oplus t$ if t can be derived by $\oplus$ summing elements of S. In the remainder of this paper, we assume that all $E_{\oplus}$ terms mentioned are in normal form, unless explicitly noted otherwise.

3 Modes of Operation

Most symmetric key ciphers are block ciphers that encrypt only fixed-length plaintext. In order to encrypt plaintexts longer than that fixed length, the encryptor divides it into a sequence of fixed-length blocks and then encrypts it using a *cryptographic mode of operation.* This is a sequence of recursively defined functions on plaintext blocks of fixed length so that each function returns a block of cipher text. To give an example, we demonstrate cipher block chaining (CBC) in Fig. 1, where the block C_0 returned by the encryptor is a random initialization vector iv, and block $C_i = E_K(m_i \oplus C_{i-1})$ for $i > 0$, where E_K is the block encryption method with key K.

We will be using part of the symbolic framework developed for the applied π-calculus [2]. In this calculus, messages exchanged in a protocol are defined over a term algebra $T_E(\Sigma, N \cup X)$, where X is a set of free variables, and N

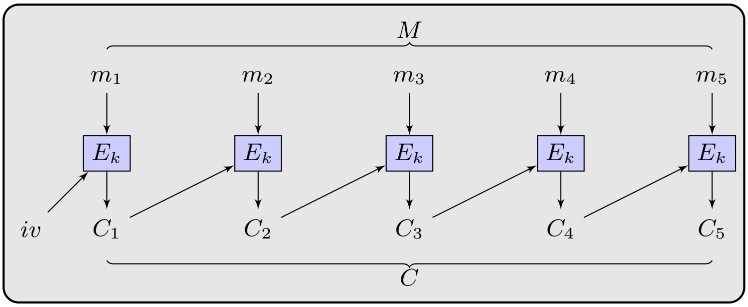

Fig. 1. An example of a cryptographic mode of operation: cipher block chaining

is a set of variables bound by the quantifier ν, standing for randomly chosen bitstrings. Protocols are defined using *processes* that describe communication between principals. A sequence of messages produced by a protocol is described using *frames*. A frame is a substitution ϕ from a set of free variables $x_1, \ldots, x_k$, to $T_E(\Sigma, N \cup X)$, i.e., $x_i\phi$ describes the i'th message sent in the frame. We may also denote a frame ϕ as $\nu R.[t_1, \ldots, t_k]$, where $t_i = x_i\phi$ and R is set of bound variables in $Ran(\phi)$.

Static equivalence in the applied π-calculus is used to describe the case in which the adversary cannot distinguish between two frames. Since all the adversary can do is combine terms via function symbols and check for equality, static equivalence is defined in terms of those actions. In our case, we have to generalize the definition slightly because, in the applied *pi* calculus it is assumed that the adversary can apply any function symbol in Σ, while in our case the adversary cannot compute f.

Definition 1. *Let $\Xi \subseteq \Sigma$. We say that two closed frames ϕ and ψ with range $T_E(\Sigma, N)$ are Ξ-statically equivalent, if $Dom(\phi) = Dom(\psi)$ and, for all terms M and N in $T_E(\Xi, N)$that share no bound variables, $M\phi =_E N\phi$ if and only if $M\psi =_E N\psi$. We say that two closed processes (that is, two processes that produce only closed frames) are Ξ-statically equivalent if any closed frame produced by one is Ξ-statically equivalent to some frame produced by the other.*

For example, consider $\Sigma_\oplus$, and $\Xi = \{\oplus, 0\}$. Then $\phi = \nu r_1.r_2.r_3[r_1, r_2, r_3]$ is Ξ-statically equivalent to $\phi = \nu s_1.s_2.s_3[s_1, s_2, s_2 \oplus s_3]$. However it is not Ξ-statically equivalent to $\phi' = \nu s_1.s_2[s_1, s_2, s_1 \oplus s_2]$, because in ψ' the third term is the exclusive-or of the first two, but the same does not hold for ϕ. Similarly, ϕ is $\{\oplus, 0\}$-equivalent to $\rho = \nu u_1.u_2[u_1, u_2, f(u_1 \oplus u_2)]$, but it is not $\Sigma_\oplus$-statically equivalent to ρ.

Note that, for the purpose of proving or disproving static equivalence, it is enough to identify processes with the sets of frames they produce. Thus, for cryptographic modes of operation that use exclusive-or, we define a $MOO_\oplus$-process as the set of closed frames that describe all possible interactions between an adversary and an encryptor in a given mode of operation.

In order to prove $\{\oplus, 0\}$-static equivalence to random, we consider frames in which each block of plaintext submitted by the adversary is denoted by a fresh free variable. We call such a frame a *symbolic history.* A mode is modeled as a $MOO_{\oplus}$-program, which describes the set of all possible symbolic histories of interaction between an adversary and an encryptor in the symbolic interaction. Each such history is a frame whose image lies in $T_{E_{\oplus}}(\Sigma_{\oplus}, N \cup X)$ where f stands for E_K (i.e., a block encryption function $E_K(m)$ with fixed key K). This frame gives in order the plaintext and ciphertext blocks exchanged between the adversary and the encryptor, where the plaintext blocks are represented by free variables. For example, the following symbolic history describes the use of the CBC mode of encryption to encrypt a two-block message: $\nu r[r, x_1, f(r \oplus x_1), x_2, f(x_2 \oplus f(r \oplus x_1))]$. Each x_i models a plaintext block sent by the adversary, and all others are terms sent by the encryptor. We also note that a symbolic history can represent the interleaving of several sessions between the adversary and an encryptor, in which a session represents the interaction between the adversary and the encryptor necessary to encrypt a single message.

The set of symbolic histories that can be produced by a mode does not by itself give us a complete description of the closed frames that can be produced by it. For that we need to specify what closed substitutions the adversary can make. For this, we need the following definition:

Definition 2. *Let H be a symbolic history, and let x be a free variable sent by the adversary in H, i.e. $H = H_1, x, H_2$ where x does not appear in any term in H_1.*

1. *We define $KN_{H,x}$ to be the set of terms in H (including free variables sent by the adversary) sent before the adversary sent x, i.e. $KN_{H,x} = \{t \mid t \in H_1\}$.*
2. *We say that $x >_H t$ if $KN_{H,x} \vdash_{\oplus} t$.*
3. *We say that a substitution σ on the free variables of H is* computable*, if for each free variable x, $x\sigma = t\sigma$ such that $t <_H x$.*

The restriction to computable substitutions captures the fact that, since the adversary cannot predict the output of f on a given input, or the choice of a random string generated by the encryptor, it can only use such outputs that it has already seen when constructing its substitutions. Note that we do not include bound variables the adversary has generated itself. Although these can be represented in the applied π calculus, they turn out not to be necessary to proving security (See Lemma 1).

Example 1. Consider the CBC mode of encryption illustrated in Fig. 1. The initial cipher block is the *iv*, $C_0 \mapsto r$ where r is a random nonce, and the second cipher block is the term $C_1 \mapsto f(x_1 \oplus C_0)$. So at this point, the symbolic history, H, contains just two blocks, C_0 and C_1, and $KH_{H,x}$ is $\{r, f(x_1 \oplus C_0)\}$. Continuing, the next block is added to H, $C_2 \mapsto f(x_2 \oplus f(x_1 \oplus C_0))$. We are able to unify C_1 and C_2 with the *computable* substitution $\sigma = \{x_1 \mapsto C_0,\ x_2 \mapsto f(0)\}$. Notice that the adversary has seen C_0 before x_1, thus $C_0 <_H x_1$ and by using this mapping can compute $f(0)$ before seeing x_2.

We now formally define a property, symbolic security, and show that it is equivalent to $\{\oplus, 0\}$-statically equivalent to random. *Symbolic security is the property we will be proving in this paper.*

Definition 3. *We say that a mode is* symbolically secure *if, for any symbolic history H, and any computable closed substitution σ on the free variables of H, there is no subset S of the set of ciphertext blocks returned by the encryptor such that $\sum_{t \in S} \oplus t =_{\oplus} 0$.*

Lemma 1. *A mode is symbolically secure if and only if it is $\{\oplus, 0\}$-statically equivalent to random.*

Proof (Sketch). Consider a mode M_{real}. Let $\phi : y_1, \ldots y_k \to T_{E_\oplus(\Sigma, N \cup X)}$. be a symbolic history from M_{real}. Let P-$Dom(\phi)$ be the set of variables in $Dom(\phi)$ mapped to variables standing for plaintext blocks, and let C-$Dom(\phi)$ be the set of variables in $Dom(\phi)$ mapped to terms standing for ciphertext blocks. Let ψ be such that $Dom(\psi) = Dom(\phi)$ and $y_i\psi$ is a fresh bound variable if $y_i \in C$-$Dom(\phi)$, and a fresh free variable if $y_i \in P$-$Dom(\phi)$. We define M_{ran} to be the mode whose symbolic histories consist of all such ψ. Thus M_{ran} is a mode in which the encryptor always returns fresh random strings.

We note that, for any computable frame $\sigma\phi$ from M_{real} there is a computable frame $\sigma\psi$ from M_{ran} constructed as above, and any computable frame from M_{ran} can be obtained this way. It is clear from the definitions that if $\sigma\phi$ and $\sigma\psi$ are $\{\oplus, 0\}$-statically equivalent then $\sigma\phi$ is symbolically secure. We now show that for any such $\sigma\phi$ and $\sigma\psi$ that, if $\sigma\phi$ is symbolically secure then $\sigma\phi$ and $\sigma\psi$ are $\{\oplus, 0\}$-statically equivalent. For that, it is enough to show that, if M and N are the exclusive-or of elements of $(Dom(\phi)$, then 1) $M\sigma\phi =_{\Sigma_\oplus} N\sigma\phi$ if and only if 2) $M\sigma\psi =_{\Sigma_\oplus} N\sigma\psi$. We note that 2) is true if and only if each ciphertext terms appears an even number of times as a summand of $M\sigma\psi \oplus N\sigma\psi$, and since by hypothesis, the ciphertext terms returned by M_{real} satisfy no nontrivial $\oplus$ equation, the same conditions apply to 1).

We now consider the case in which the adversary may include bound variables, generated by itself, as summands of the plaintext. In the applied π calculus this is done by prepending to the frame the sequence of bound variables generated by the adversary. It is then straightforward to reduce this to the computable case with no adversary-generated bound variables. □

Given a cryptographic mode of operation, we can define several instances of the security problem, based on combination of different factors. These include the schedule (e.g. are ciphertext blocks returned by the encryptor only after all plaintext blocks are received (messagewise schedule), or as soon as the encryptor can compute them (blockwise schedule)), and the bounds on session length and number of sessions. We will use the following modes of operation as examples.

- Cipher Block Chaining (CBC) : The i^{th} plain text is a ground $MOO_\oplus$-term p_i. The initial cipher block, the iv, is modeled by a bound variable, r. The i^{th} block of cipher text, C_i, is modeled by the term $f(C_{i-1} \oplus p_i)$. This is secure in the messagewise schedule, but not in the blockwise.

- Cipher Feedback (CFB) : The i^{th} plain text is modeled by a ground $MOO_{\oplus}$ term p_i. The initial cipher block, the iv, is modeled by a bound variable, r. The i^{th} cipher block, C_i, is modeled by the term $f(C_{i-1}) \oplus p_i$. This is secure under both schedules.
- Similarly, Propagating Cipher Block Chaining $(PCBC)$: $C_1 = f(p_1 \oplus IV)$, $C_i = f(p_i \oplus p_{i-1} \oplus C_{i-1})$. This is secure under the messagewise schedule but not under the blockwise schedule.

4 The Invertibility Problem

A natural requirement of any cryptographic algorithm is that it be *invertible*; that is, one can find the original plaintext using the ciphertext and decryption key. While this property would normally be "built-in" to a mode of operation, it is not guaranteed to exist for all possible modes that can be automatically generated, even if these modes have other desirable properties such as symbolic security. Therefore in the automatically generated setting, we will need methods for checking if the invertibility property holds for any particular $MOO_{\oplus}$-program. This leads to two different questions.

- The first is, given a set S of $MOO_{\oplus}$-terms with subterms designated as plain text, can we tell if S is invertible? This bounded version of the problem follows from the Abadi and Cortier's [1] results on the decidability of deducability in various equational theories and
- The second is: given a $MOO_{\oplus}$-program, can we tell if an arbitrary cipher block is invertible? We explore this un-bounded version of the problem in this section.

Let $\mathcal{C} = \{C_0, C_1, \ldots, C_n\}$ represent the ciphertext blocks, C_i, produced by the encryptor in the $MOO_{\oplus}$-program. We instantiate the variables representing plaintext in $\mathcal{C}$ to bound variables p_i. Let $P = \{p_0, p_1, \ldots, p_n\}$ be the set representing the plaintext messages during a run of the $MOO_{\oplus}$-program. We introduce a new symbol, f^{-1}, where f is the symbolic encryption function, i.e., $f = enc(\_, K)$, for some key K, and let f^{-1} model decryption, $f^{-1} = dec(\_, K)$, s.t. $f^{-1}(f(M)) = M$. Then $E^{-1} = E_{\oplus} \cup \{f^{-1}(f(x)) = x\}$.

Lemma 2. *Let t be a closed term over $f, \oplus, 0$ and let $c \in fn(t)$. Let S be a set of terms consisting of t and every bound variable in t other than c. Then c can be deduced from S if and only if c appears exactly once in t.*

Proof. We first prove the "if" part. If $|t| = 1$, then $t = c$. Assume c is deducible for terms whose size is k or less. When size $|t| = k + 1$, the term either contains an $\oplus$ or f at the root, i.e., either $t = f(t')$ for some t', or $t = t_1 \oplus t_2$ for some t_1, t_2 where $t_1 \oplus t_2$ cannot be further simplified. When $t = f(t')$, we remove the f symbol by applying f^{-1}. Then $|t'| = k$, and t' contains c. By the induction hypothesis c can be deduced for terms up to size k, i.e., from set S. When $t = t_1 \oplus t_2$, without loss of generality we can assume that c appears exactly once in t_1, thus t_2 is known. The size of $t_1 \leq k$ and by induction hypothesis c can be derived from t_1. The "only if" part follows from the fact that given a known term $t_1 \oplus t_2$, neither t_1 nor t_2 can be deduced from it unless one of t_1 or $t_2 \in S$. □

Definition 4. *Consider recursive definitions which satisfy the following restrictions:*

1. *The base case, C_0, is the initial random nonce and the only nonce, i.e., a bound variable that is computed by the encryptor.*
2. *C_i contains the i^{th} plaintext p_i, represented by a bound variable.*
3. *p_i appears only once in C_i.*

Directly from Lemma 2 and Definition 4 we obtain the following.

Theorem 1. *Cryptosystems defined using Definition 4 are invertible, i.e., for all $i \geq 0$, p_i can be deduced from $\{C_0, \ldots, C_i\}$.*

4.1 Implementation

Invertibility has been implemented via an algorithm based on Theorem 1. The algorithm is restricted to the set of $MOO_{\oplus}$-programs of Definition 4. The benefit of this algorithm is that it doesn't require the production of actual $MOO_{\oplus}$-terms, but can be applied directly to the recursive definition of the cryptosystem.

5 Decision Problems for Symbolic Security

In this section we prove results concerning decidability of symbolic security. For this we concentrate on modes of operation in which ciphertext blocks are of the form $x \oplus G$, where x is a free variable, and G contains no free variables. For such a mode, proving symbolic security reduces to proving that there is no symbolic history H containing a sequence $x_1 \oplus G_1, \ldots, x_k \oplus G_k$ such that $\sum_{i=1}^{k} \oplus G_i =_{E_{\oplus}} 0$. It is interesting to note that the problem is undecidable even when G_i contains no free variables, which means deciding it only requires checking for equality, not performing unification. Indeed, not only is the problem undecidable, but it is undecidable even when we bound some of the parameters, e.g. the number or length of sessions. We use an approach similar to that of Küsters and Truderung in [7], in which the security of recursive protocols defined in a term algebra that is a superset of ours is shown to be undecidable.

5.1 Undecidable Decision Problems for Block Ciphers

Due to space we consider just one type of decision problem here, those with sessions of an arbitrary or unbounded length but for which the number of sessions is bounded. That is, we do not assume a bound on the length of the interaction between the adversary and encryptor. However, we do assume a finite bound on the number of possible interleaved sessions the adversary may create. In fact, since a single session is sufficient to obtain the undecidability results we will just consider that case.

There are then two sub-cases of these unbounded length but bounded number of sessions problems. The cases are based on whether the $MOO_{\oplus}$-program is modeled by a non-deterministic function or a deterministic function. In this section we examine the non-deterministic case, where a session may have non-deterministic choice points, and the adversary chooses which path is taken. The second, deterministic case, and several additional related problems can be proven in a similar manner.

Definition 5. *Let α be a string $a_0a_1\ldots a_m$ and let C be a block. Then, $F(\alpha \bigoplus C) = f(a_0 \oplus f(a_1 \oplus \ldots f(a_m \oplus C)\ldots))$.*

We will use the following method for constructing ciphertext blocks. The construction encodes possible solutions to the Post Correspondence Problem (PCP).

Definition 6. *Let $PCP = (\frac{\alpha_0}{\beta_0}),\ (\frac{\alpha_1}{\beta_1}),\ \ldots,\ (\frac{\alpha_n}{\beta_n})$. Let $L = j_0, j_1, \ldots, j_k$ be a sequence of integers such that $0 \leq j_i \leq n$, and let $L_i = [j_{k-i}, \ldots, j_k]$. (Thus, $L_0 = [j_k]$ and $L_k = L$.) For $k \geq i > 0$ let $E_{i,L_i} = [f(r_i \oplus C_{i,L_i,1}) \oplus x_{i,1}, f(r_i \oplus C_{i,L_i,2}) \oplus x_{i,2}]$, $0 \leq j \leq n$, $C_{i,L_i,1} = F(\alpha_{j_i} \bigoplus C_{i-1,L_{i-1},1})$, and $C_{i,L_i,2} = F(\beta_{j_i} \bigoplus C_{i-1,L_{i-1},2})$. Where each r_i is a fresh bound variable, and $C_{0,L_0,1} = F(\alpha_{j_0} \bigoplus 0)$, $C_{0,L_0,2} = F(\beta_{j_0} \bigoplus 0)$.*

Essentially, the definition encodes any sequence of PCP blocks. Each E_{i,L_i} contains two strings. The first string encodes a sequence of α strings, from the tops of the PCP blocks, and the second string encodes a sequence of β strings, from the bottoms of the PCP blocks. Thus, any solution to the PCP problem can be encoded into a sequence of mode of encryption style cipher blocks (see Example 2).

Based on the non-deterministic system of Definition 6 we can define the following $MOO_{\oplus}$-program, which produces two equal cipherblocks which sum to zero iff the adversary finds a solution to the PCP.

Definition 7. *Denote the following $MOO_{\oplus}$-program as PCP_{NDMOO_1}. The program works as follows:*

- *The adversary non-deterministically picks a possible solution to the PCP, $[L = j_0, j_1, j_2, \ldots, j_k]$.*
- *At the adversary's i^{th} turn, it sends index j_{k-i} of the solution to the encryptor, as well as two plaintext blocks, $x_{i,1}$ and $x_{i,2}$.*
- *At i^{th} step the encryptor's i^{th} turn encodes a pair of ciphertext blocks $E_{i,L_i} = [f(r_i \oplus C_{i,L_i,1}) \oplus x_{i,1}, f(r_i \oplus C_{i,L_i,2}) \oplus x_{i,2}]$, according to Definition 6 and returns the pair to the adversary.*
- *After receiving each E_{i,L_i}, the adversary sums $f(r_i \oplus C_{i,L_i,1}) \oplus x_{i,1}$ with $x_{i,1}$ and $f(r_i \oplus C_{i,L_i,2}) \oplus x_{i,2}$ with $x_{i,2}$ to obtain the blocks $f(r_i \oplus C_{i,L_i,1})$ and $f(r_i \oplus C_{i,L_i,2})$.*

- *The program stops if $f(r_i \oplus C_{i,L_i,1}) = f(r_i \oplus C_{i,L_i,2})$ or the adversary stops sending input to the encryptor.*

Example 2. Consider the following PCP:

$$\overbrace{\left(\frac{ba}{baa}\right)}^{\text{tile 1}}, \overbrace{\left(\frac{ab}{ba}\right)}^{\text{tile 2}}, \overbrace{\left(\frac{aaa}{aa}\right)}^{\text{tile 3}}$$

One solution to this problem is $[1, 3]$. Let's trace a run of the $MOO_{\oplus}$-program PCP_{NDMOO_1} where the adversary guesses the solution $[1, 3]$. In the first step the adversary sends 3 to the encryptor and receives: $E_{0,[3]} = [f(r_0 \oplus C_{0,[3]1}) \oplus x_{0,1}, f(r_0 \oplus C_{0,[3],2}) \oplus x_{0,2}]$, $C_{0,[3],1} = F(\alpha_3 \bigoplus 0) = (f(a \oplus f(a \oplus f(a \oplus 0))))$, $C_{0,[3],2} = F(\beta_3 \bigoplus 0) = f(a \oplus f(a \oplus 0))$.

At the second step the adversary sends a 1 to the encryptor and receives the following in return. $E_{1,[1,3]} = [f(r_1 \oplus C_{1,[1,3],1}) \oplus x_{1,1}, f(r_1, C_{1,[1,3],2}) \oplus x_{1,2}]$, $C_{1,[1,3],2} = F(\alpha_1 \bigoplus C_{0,[3],1}) = f(b \oplus f(a \oplus C_{0,[3],1}))$, $C_{1,[1,3],2} = F(\beta_1 \bigoplus C_{0,[3],2}) = f(b \oplus f(a \oplus f(a \oplus C_{0,[3],2})))$.

Notice that now after step 2 the adversary has two ciphertext blocks, $C_{1,[1,3],1}$ and $C_{1,[1,3].2}$, which are equal and therefore their sum will be equal to zero. $C_{1,[1,3],1} = f(b \oplus f(a \oplus f(a \oplus f(a \oplus f(a \oplus 0)))))$, $C_{1,[1,3],2} = f(b \oplus f(a \oplus f(a \oplus f(a \oplus f(a \oplus 0)))))$.

Lemma 3. *A given PCP problem has a solution if and only if there is a sequence L of indices of that problem such that the $MOO_{\oplus}$-program PCP_{NDMOO_1} is symbolically secure.*

Proof (Sketch). Since each block returned by the encryptor is the sum of an f-rooted term and a free variable, symbolic security is violated if and only if two of these f-rooted terms are unified. Assume that two such terms are found to be equal. Due to the random r_i at each step the only blocks that are possibly equal are blocks from the same step, C_{i,L_i1} and $C_{i,L_i,2}$. If these blocks are equal then there is a solution to the PCP. Conversely, suppose that $[i_1, i_2, \ldots, i_m]$ is a solution to the PCP. Notice that during the *mth* step that the blocks $C_{m,[],1}$ and $C_{m,[],2}$ will fully encode this solution. □

Directly from Lemma 3 we obtain the following.

Theorem 2. *Assume M is an arbitrary non-deterministic $MOO_{\oplus}$-program. The problem of determining if M, executing with a bounded number of sessions and unbounded session lengths, is symbolically secure is undecidable.*

Several additional undecidability results can be proven using a similar reduction. These cases include deterministic unbounded session length, both deterministic and non-deterministic unbounded number of sessions with bounded session length.

5.2 An Algorithm for Checking Symbolic Security

While the question of symbolic security of modes of operation is undecidable in general, this section explores a sufficient condition for symbolic security, and gives an algorithm for checking symbolic security of modes of operation.

Let M be any mode of operation. Let H be a symbolic history of M, which can be an interleaving of multiple sessions, each of which is used to encrypt a single message of some plaintext blocks. M is defined inductively as $₵_{p,i} = t_{ind}, ₵_{p,0} = t_0$. We call $₵_{p,i}$ a *ciphertext variable*, and use it to denote the i^{th} ciphertext block from the p^{th} session. We call $x_{p,i}$ a *plaintext variable*, and use it to denote the i^{th} plaintext block from the p^{th} session. If we *unfold* $₵_{p,i}$, we get t_{ind}. We assume that t_{ind} is a $MOO_{\oplus}$-term of the form $f(t_1) \oplus \ldots \oplus f(t_m) \oplus x_{p,i}$. We use *top-$f$-terms*$(₵_{p,i})$ to denote $\{f(t_1), \ldots, f(t_m)\}$. Each $f(t_j)$ $(1 \leq j \leq m)$ is called an *f-rooted summand* of $₵_{p,i}$. We define $size_f(₵_{p,i})$ to be the number of f-rooted summands of $₵_{p,i}$.

Let t_1 and t_2 be two $MOO_{\oplus}$-terms. If $t_1\sigma =_{\oplus} t_2\sigma$, then we say that t_1 and t_2 are *$\oplus$-unifiable* under σ, or $\{t_1 \overset{?}{=} t_2\}$ is *$\oplus$-unifiable* under σ. Let Γ be a set of equations. If each equation in Γ is $\oplus$-unifiable under σ, then we say that Γ is $\oplus$-unifiable under σ.

Example 3. We use M_{CBC} to denote Cipher Feedback Mode, where
$₵_{p,i} = f(₵_{p,i-1}) \oplus x_{p,i}$, $₵_{p,0} = r_p$
(1) Here is a possible symbolic history of M_{CBC}:
$H = [r_1, r_2, x_{1,1}, f(r_1) \oplus x_{1,1}, x_{2,1}, f(r_2) \oplus x_{2,1}, x_{1,2}, f(f(r_1) \oplus x_{1,1}) \oplus x_{1,2}]$.
(2) Here is a computable substitution on H:
$\sigma = \{x_{1,1} \mapsto 0, x_{2,1} \mapsto f(r_1), x_{1,2} \mapsto f(r_1) \oplus r_2\}$.
$H\sigma = [r_1, r_2, 0, f(r_1), f(r_1), f(r_2) \oplus f(r_1), f(r_1) \oplus r_2, f(f(r_1)) \oplus f(r_1) \oplus r_2]$.

Note that, in the above example, there are no ciphertext blocks in $H\sigma$ such that they sum to 0. Here is the intuition. Let S be the set of all f-rooted summands of $MOO_{\oplus}$-terms in H. So $S = \{f(r_1), f(r_2), f(f(r_1) \oplus x_{1,1})\}$. No two $MOO_{\oplus}$-terms in S are unifiable under any computable substitution of H. We formalize this observation using the following Definition 8.

Definition 8. *Let M be a mode of operation. Consider any symbolic history H of M. Let $₵_{p,i}$ and $₵_{q,j}$ be any two ciphertext blocks in H. M satisfies the uniqueness property if for any two distinct $MOO_{\oplus}$-terms $t_1, t_2 \in$ top-f-terms$(₵_{p,i}) \cup$ top-f-terms$(₵_{p,j})$, there does not exist any computable substitution σ of H s.t. $t_1\sigma =_{\oplus} t_2\sigma$.*

The following lemma states that the uniqueness property implies symbolic security.

Lemma 4. *Let M be any mode of operation. If M satisfies the uniqueness property, then M is symbolically secure.*

Proof. Let M be a mode of operation. Consider any symbolic history H of M and any computable substitution σ. Let $S : ₵_{p_1,i_1}, \ldots, ₵_{p_m,i_m}$ be a subsequence of H. By the uniqueness property, $\sum_{k=1}^{m} \oplus \mathbb{C}_{p_k,i_k}\sigma = \mathit{top\text{-}f\text{-}terms}(\mathbb{C}_{p_m,i_m})\sigma \oplus t$ for some t. □

Let M be a mode of operation, H be any symbolic history of M. The following Definition 9 defines the notion of a *crucial pair* of H. Intuitively, a crucial pair is the earliest unifiable pair of f-rooted $MOO_{\oplus}$-terms in H. In order to show that M satisfies the uniqueness property, we show that M does not admit any symbolic history, where a crucial pair exists.

$$\frac{\Gamma \cup \{f(t) \overset{?}{=} 0\}}{\Gamma}\ \mathit{Elim}_f$$

$$\frac{\Gamma \cup \{\mathbb{C}_{p,m} \oplus \mathbb{C}_{q,n} \overset{?}{=} 0\}}{\Gamma}\ \mathit{Elim}_C$$

where $i \neq j$ implies $m \neq n$.

$$\frac{\Gamma \cup \{\mathbb{C}_{p,m} \oplus f(t) \overset{?}{=} 0\}}{\Gamma}\ \mathit{Occurs\_check}$$

where $\mathbb{C}_{p,m}$ is a subterm of t.

$$\frac{\Gamma \cup \{f(t_1) \oplus \ldots \oplus f(t_n) \overset{?}{=} 0\}}{\Gamma \cup \{t_k \oplus t_1 \overset{?}{=} 0\} \cup \ldots \{t_k \oplus t_{k-1} \overset{?}{=} 0\} \cup \{t_k \oplus t_{k+1} \overset{?}{=} 0\} \cup \ldots \cup \{t_k \oplus t_n \overset{?}{=} 0\}}\ \mathit{Pick}_f$$

where k is chosen nondeterministically between 1 and n.

$$\frac{\Gamma \cup \{\mathbb{C}_{p,m} \oplus f(t_1) \oplus \ldots \oplus f(t_n) \overset{?}{=} 0\}}{\Gamma \cup \{t'_u \overset{?}{=} t_1\} \cup \ldots \{t'_u \overset{?}{=} t_n\}}\ \mathit{Pick}_C$$

where (1) $f(t'_u)$ is an f-rooted summand of $\mathbb{C}_{p,m}$. (2) $size_f(\mathbb{C}_{p,m}) \leq n$. (3) $\mathbb{C}_{p,m} \in C\_Var(tm) \cup C\_Var(tm')$.

$$\frac{\Gamma \cup \{\mathbb{C}_{p,m} \oplus f(t_1) \oplus \ldots \oplus f(t_n) \overset{?}{=} 0\}}{\Gamma}\ \mathit{Pick}_{fail}$$

where $size_f(\mathbb{C}_{p,m}) > n$.

Fig. 2. Inference system $\mathcal{I}_{i,j,tm,tm'}$

Definition 9. *Let M be a mode of operation, H be any symbolic history of M.*

(1) Suppose that t_1 is an f-rooted summand of $\mathbb{C}_{p,i}$, t_2 is an f-rooted summand of $\mathbb{C}_{q,j}$.
- *If $\mathbb{C}_{p,i}$ appears no later than $\mathbb{C}_{q,j}$ in H, then $t_1 \preceq t_2$.*
- *If $\mathbb{C}_{p,i}$ appears earlier than $\mathbb{C}_{q,j}$ in H, then $t_1 \prec t_2$.*

(2) t_1 and t_2 are a crucial pair of H w.r.t (i, j, σ) if
- *There exist some $\mathbb{C}_{p,i}$ and $\mathbb{C}_{q,j}$ in H s.t. t_1 is an f-rooted summand of $\mathbb{C}_{p,i}$, t_2 is an f-rooted summand of $\mathbb{C}_{q,j}$.*
- *σ is a computable substitution of H, and $t_1\sigma =_{\oplus} t_2\sigma$.*

- *If $t_1' \prec t_1$ and $t_2' \preceq t_2$, or $t_1' \preceq t_1$ and $t_2' \prec t_2$, then for any computable substitution σ of H, $t_1'\sigma \neq_{\oplus} t_2'\sigma$.*

In order to show that no crucial pair exists in a symbolic history H, we take any two ciphertext blocks $\mathbb{C}_{p,i}$ and $\mathbb{C}_{q,j}$ in H. We then consider two different f-rooted summands tm and tm' of $\mathbb{C}_{p,i}$ and $\mathbb{C}_{q,j}$. We assume that tm and tm' are a crucial pair of H, and try to derive a contradiction using the inference rules in $\mathcal{I}_{i,j,tm,tm'}$ (Fig. 2), starting from an initial set of equations $\{tm \oplus tm' \overset{?}{=} 0\}$. Note that $\mathcal{I}_{i,j,tm,tm'}$ is parameterized by i, j, tm and tm', which are referred to by $Elim_C$ and $Pick_C$. We use $\mathcal{I}^k_{i,j,tm,tm'}(\{tm \oplus tm' \overset{?}{=} 0\})$ to represent the set of equations that we get after the k^{th} inference step. We use $\mathcal{I}_{i,j,tm,tm'}(\{tm \oplus tm' \overset{?}{=} 0\})$ to represent the final result. We maintain the following invariant: If we get a set of equations Γ at any step, and tm and tm' are unifiable under some computable substitution, then at least one of the equations in Γ must hold. Intuitively, each equation in Γ represents a possibility that tm and tm' are unifiable under a computable substitution, and Γ represents the set of all possibilities. Our goal is to derive a contradiction, which is to make Γ empty.

The $Elim_f$ rule allows us to remove the possibility that an f-rooted $MOO_{\oplus}$-term is 0. The $Elim_C$ rule allows us to remove the possibility that we somehow find an earlier pair of unifiable terms. Unification of $\mathbb{C}_{p,m}$ with a $MOO_{\oplus}$-term strictly containing it is impossible by the $Occurs\_check$ rule. If the xor of some f-rooted terms is 0, the $Pick_f$ rule nondeterministically picks one of them and list all the possibilities that it can cancel with some other f-rooted $MOO_{\oplus}$-term. If the number of f-rooted summands of $\mathbb{C}_{p,m}$ is greater than the number of f-rooted terms in an equation, the $Pick_{fail}$ rule applies. The $Pick_C$ rule first unfolds $\mathbb{C}_{p,m}$, then picks an f-rooted summand of $\mathbb{C}_{p,m}$ and cancels it with some f-rooted term. Note that the $Pick_C$ rule rules out the possibility that two f-rooted summands of $\mathbb{C}_{p,m}$ can cancel with each other. In order to apply the $Pick_C$ rule, $\mathbb{C}_{p,m}$ must be a ciphertext variable of either tm or tm'. We need this condition for termination.

Algorithm 1. Checking Symbolic Security of Modes of Operation

```
Input: a recursive description of some mode of operation $M$.

$\Gamma = top\text{-}f\text{-}terms(\mathbb{C}_{p,i}) \cup top\text{-}f\text{-}terms(\mathbb{C}_{q,j})$
for each pair of distinct terms $tm$ and $tm'$ in $\Gamma$ do
  if $\mathcal{I}_{i,j,tm,tm'}(\{tm \oplus tm' \overset{?}{=} 0\}) \neq \emptyset$ then
    return "unknown"
  end if
end for
return "secure"
```

Definition 10. *Given a $MOO_{\oplus}$-term t, $C\_Var(t)$ denotes the set of ciphertext variables occurring in t. More formally,* *(1) $C\_Var(\mathcal{C}_{p,i}) = \{\mathcal{C}_{p,i}\}$, if $\mathcal{C}_{p,i}$ is a ciphertext variable. (2) $C\_Var(x_{p,i}) = \emptyset$, if $x_{p,i}$ is a plaintext variable. (3) $C\_Var(f(t)) = C\_Var(t)$. (4) $C\_Var(t_1 \oplus t_2) = C\_Var(t_1) \cup C\_Var(t_2)$.*

The following Lemma 5 describes an important invariant of $\mathcal{I}_{i,j,tm,tm'}$, which implies the soundness of Algorithm 1.

Lemma 5. *Let M be a mode of operation, H be any symbolic history of M. Suppose that tm and tm' are a crucial pair of H w.r.t. (i, j, σ). For all k, if $\mathcal{I}^k_{i,j,tm,tm'}(\{tm \oplus tm' \overset{?}{=} 0\}) = \Gamma$, at least one equation in Γ must be $\oplus$-unifiable under σ.*

Proof (Sketch). We prove this lemma by induction on k. When $k = 0$, the lemma holds trivially. Assume that the lemma holds when $k = l - 1$. We want to show that the lemma also holds when $k = l$. Consider the l^{th} inference step.

If $Elim_f$, $Elim_C$, $Occurs\_check$ or $Pick_{fail}$ is used, an impossible case is removed. For example, if $Elim_C$ is used, $\{\mathcal{C}_{p,m} \oplus \mathcal{C}_{q,n} \overset{?}{=} 0\}$ is impossible, since it contradicts with the assumption that tm and tm' are a crucial pair of H w.r.t. (i, j, σ). If $Pick_f$ or $Pick_C$ is used, we nondeterministically guess an f-rooted term and list all the possibilities that it can cancel with some other term. □

Theorem 3 (Soundness). *For any mode of operation M, if Algorithm 1 returns "secure", then M is symbolically secure.*

Proof. Given a mode of operation M, if Algorithm 1 returns "secure", then for each pair of distinct terms tm and tm' in $top\text{-}f\text{-}terms(\mathcal{C}_{p,i}) \cup top\text{-}f\text{-}terms(\mathcal{C}_{q,j})$, $\mathcal{I}_{i,j,tm,tm'}(\{tm \oplus tm' \overset{?}{=} 0\}) = \emptyset$. By Lemma 5, no pair of terms tm and tm' are a crucial pair of H. This means that the uniqueness property holds for M. Therefore, by Lemma 4, M is symbolically secure. □

To prove termination of Algorithm 1, we define the following relations: $\prec_E$ and $\preceq_E$ are partial order relations on equations, $\prec_S$ is a partial order on sets of equations.

Definition 11. *Let eq be an equation of the form $t_1 \oplus \ldots \oplus t_m \overset{?}{=} 0$, where each t_i $(1 \leq i \leq m)$ is either f-rooted or a bound variable. Let eq' be an equation of the form $t'_1 \oplus \ldots \oplus t'_n \overset{?}{=} 0$, where each t'_i $(1 \leq i \leq n)$ is either f-rooted or a bound variable. We say that $eq \prec_E eq'$ if for all $1 \leq i \leq m$, there exists j such that t_i is a strict subterm of t'_j. We say that $eq \preceq_E eq'$ if $eq \prec_E eq'$ or eq is the same as eq'.*

Let $\Gamma = \{eq_1, \ldots, eq_m\}$, $\Gamma' = \{eq'_1, \ldots, eq'_n\}$. We say that $\Gamma \prec_S \Gamma'$ if for all $1 \leq i \leq m$, there exists j such that $eq_i \preceq_E eq'_j$, and at least one of the following conditions is true: (1) $|\Gamma| < |\Gamma'|$. (2) There exists i, j, s.t. $eq_i \prec_E eq'_j$.

Let Γ be a set of equations, let t and t' be two $MOO_\oplus$-terms. We define the following set. $C\_Var_{t,t'}(\Gamma)$ is the set of ciphertext variables that must occur in Γ, and also occur in either t or t'.

$C\_Var_{t,t'}(\Gamma) = \{\mathcal{C}_{u,v} \mid \mathcal{C}_{u,v} \in C\_Var(t) \cup C\_Var(t'), \mathcal{C}_{u,v} \text{ occurs in } \Gamma\}$.

Theorem 4 (Termination). *For any mode of operation M, Algorithm 1 always terminates.*

Proof. We show that for each tm and tm', $\mathcal{I}_{i,j,tm,tm'}$ always terminates. Consider some inference step. Suppose that we apply $\mathcal{I}_{i,j,tm,tm'}$ to Γ and get Γ'. There are 2 cases to consider.

Case 1: If $Pick_C$ is used, $|C\_Var_{tm,tm'}(\Gamma')| < |C\_Var_{tm,tm'}(\Gamma)|$.

Case 2: If $Elim_f, Elim_C, Occurs\_check, Pick_f$ or $Pick_{fail}$ is used, then $|C\_Var_{tm,tm'}(\Gamma')| = |C\_Var_{tm,tm'}(\Gamma)|$ and $\Gamma' \prec_S \Gamma$.

So either $|C\_Var_{tm,tm'}(\Gamma')| < |C\_Var_{tm,tm'}(\Gamma)|$, or $|C\_Var_{tm,tm'}(\Gamma')| = |C\_Var_{t,t'}(\Gamma)|$ and $\Gamma' \prec_S \Gamma$. For each tm and tm', $\mathcal{I}_{i,j,tm,tm'}$ always terminates. Therefore, Algorithm 1 always terminates. □

Here is an example of checking symbolic security using Algorithm 1.

Example 4. Let M be Cipher Feedback Mode, where: $\mathbb{C}_{p,i} = f(\mathbb{C}_{p,i-1}) \oplus x_{p,i}$, $\mathbb{C}_{p,0} = r_p$. According to Algorithm 1, $\Gamma = \{f(\mathbb{C}_{p,i-1}), f(\mathbb{C}_{q,j-1})\}$. Apply the inference system $\mathcal{I}_{i,j,f(\mathbb{C}_{p,i-1}),f(\mathbb{C}_{q,j-1})}$ to $\{f(\mathbb{C}_{p,i-1}) \oplus f(\mathbb{C}_{q,j-1}) \overset{?}{=} 0\}$.

$$\frac{\{f(\mathbb{C}_{p,i-1}) \oplus f(\mathbb{C}_{q,j-1}) \overset{?}{=} 0\}}{\{\mathbb{C}_{p,i-1} \oplus \mathbb{C}_{q,j-1} \overset{?}{=} 0\}}\ Pickf \qquad \frac{\{\mathbb{C}_{p,i-1} \oplus \mathbb{C}_{q,j-1} \overset{?}{=} 0\}}{\emptyset}\ Elim_C$$

Algorithm 1 returns “secure”.

5.3 Implementation

The CryptoSolve tool can check for symbolic-security in several ways. The first, and most exhaustive, is via the P-unification approach [8]. In this approach, cipher blocks of the MOO-program under consideration are generated and the appropriate P-unification is used to check security (see [11]). The difficulty with this approach is that it can be time consuming in practice, due to the need to continually generate, then check new cipher blocks. However, the algorithm specified in Sect. 5.2 doesn't require the explicit generation of cipher blocks, but only requires us to compare. This approach is not complete but works for many cases. Thus, we are implementing it as a first pass symbolic security check.

6 Conclusions

We have investigated two algorithmic problems arising from the symbolic analysis of cryptographic modes of operation built using block ciphers and exclusive-or: symbolic security and invertibility. We have given algorithmic results for both. We also believe that we have learned something from treating the problems separately from each other. For example, one might ask if the restrictions imposed by invertibility might narrow the class of cryptosystems to ones for which IND$-security is decidable. Our results on undecidability of symbolic show that they do not, because our embedding of the Post Correspondence Problem all produce invertible cryptosystems.

There are many ways these results can be extended. We can, as mentioned in the introduction, investigate algorithms for deciding combinations of properties. We can investigate larger classes of modes that use additional primitives and functions, such as hash functions, field operations, concatenation, block ciphers with tweaks, and the successor function, the latter two of which have already been studied in [5,9] for the messagewise schedule. In addition, we can investigate other classes of modes built using the same or similar primitives, e.g. hash functions (studied in [10]), hash-based signatures, garbled circuits (studied in [4]), and message authentication codes (studied in [5]). We also intend to determine what other cryptosystems or classes of cryptosystems are amenable to symbolic analyses and study them if feasible.

References

1. Abadi, M., Cortier, V.: Deciding knowledge in security protocols under equational theories. Theoret. Comput. Sci. **367**(1–2), 2–32 (2006)
2. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2001, pp. 104–115. ACM, New York (2001). https://doi.org/10.1145/360204.360213
3. Borgström, J.: Static equivalence is harder than knowledge. In: Baeten, J.C.M., Phillips, I.C.C. (eds.) Proceedings of the 12th Workshop on Expressiveness on Concurrency, EXPRESS 2005, San Francisco, CA, USA, 27 August 2005, pp. 45–57. Electronic Notes in Theoretical Computer Science, Elsevier (2005). https://doi.org/10.1016/j.entcs.2006.05.006
4. Carmer, B., Rosulek, M.: Linicrypt: a model for practical cryptography. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 416–445. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_15
5. Hoang, V.T., Katz, J., Malozemoff, A.J.: Automated analysis and synthesis of authenticated encryption schemes. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 84–95. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2810103.2813636
6. Joux, A., Martinet, G., Valette, F.: Blockwise-adaptive attackers revisiting the (in)security of some provably secure encryption modes: CBC, GEM, IACBC. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 17–30. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_2
7. Küsters, R., Truderung, T.: On the automatic analysis of recursive security protocols with XOR. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, pp. 646–657. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70918-3_55
8. Lin, H., Lynch, C.: Local XOR unification: definitions, algorithms and application to cryptography. IACR Cryptol. ePrint Arch. 2020, 929 (2020). https://eprint.iacr.org/2020/929
9. Malozemoff, A.J., Katz, J., Green, M.D.: Automated analysis and synthesis of block-cipher modes of operation. In: 2014 IEEE 27th Conference on Computer Security Foundations Symposium (CSF), pp. 140–152. IEEE (2014)

10. McQuoid, I., Swope, T., Rosulek, M.: Characterizing collision and second-preimage resistance in Linicrypt. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. LNCS, vol. 11891, pp. 451–470. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_18
11. Meadows, C.A.: Symbolic and computational reasoning about cryptographic modes of operation. IACR Cryptol. ePrint Arch. 2020, 794 (2020). https://eprint.iacr.org/2020/794
12. Rogaway, P.: Nonce-based symmetric encryption. In: 11th International Workshop on Fast Software Encryption, FSE 2004, Delhi, India, 5–7 February 2004, Revised Papers, pp. 348–359 (2004). https://doi.org/10.1007/978-3-540-25937-4_22