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

Hai Lin [1] Cristopher Lynch [1]
Andrew M. Marshall [2]
Catherine A. Meadows [3]
Paliath Narendran [4] Veena Ravishankar [2]
Brandon Rozek [2]

[1]Clarkson University, Potsdam, NY, USA

[2]University of Mary Washington, Fredericksburg, VA, USA

[3]Naval Research Laboratory, Washington, DC, USA

[4]University at Albany–SUNY, Albany, NY, USA

Introduction

Hello and welcome to this talk!

Introduction

The goals of this talk are the following:

- Give a brief overview of recent, and I think, interesting work on applying automated reasoning to cryptography. In particular to the automatic generation and security proofs of cryptographic systems.
- Explain some of the interesting problems that results such as a new type of unification problem.
- Introduce a new tool which is an implementation of some of these methods.

Background

First, a very quick look at some background material

Block Cipher Mode of Operation

In this talk we will just consider symmetric key block ciphers.

It's often useful to talk about these schemes as "modes of operations"

Block Cipher Mode 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 *block cipher mode of operation* (MOO).

This is a sequence of recursively defined functions on plaintext blocks of fixed length so that each function returns a block of cipher text.

MOO Example: ECB

Definition 1

MOO: ECB

Let E_k be a block cipher with block length n. Let M be the message such that $M = m_1 \cdot m_2, \ \ldots \ \cdot m_l$ such that $m_i \in \{0,1\}^n$. Let C_i, $1 \leq i \leq l$, be n length cipher blocks such that:

$$C_i = E_k(m_i)$$

Then, the ciphertext $C = C_1 \cdot C_2 \cdot \ \ldots \ \cdot C_l$

MOO Example: ECB

Here is a pictorial example of a very simple mode, Electronic Code Book (ECB).

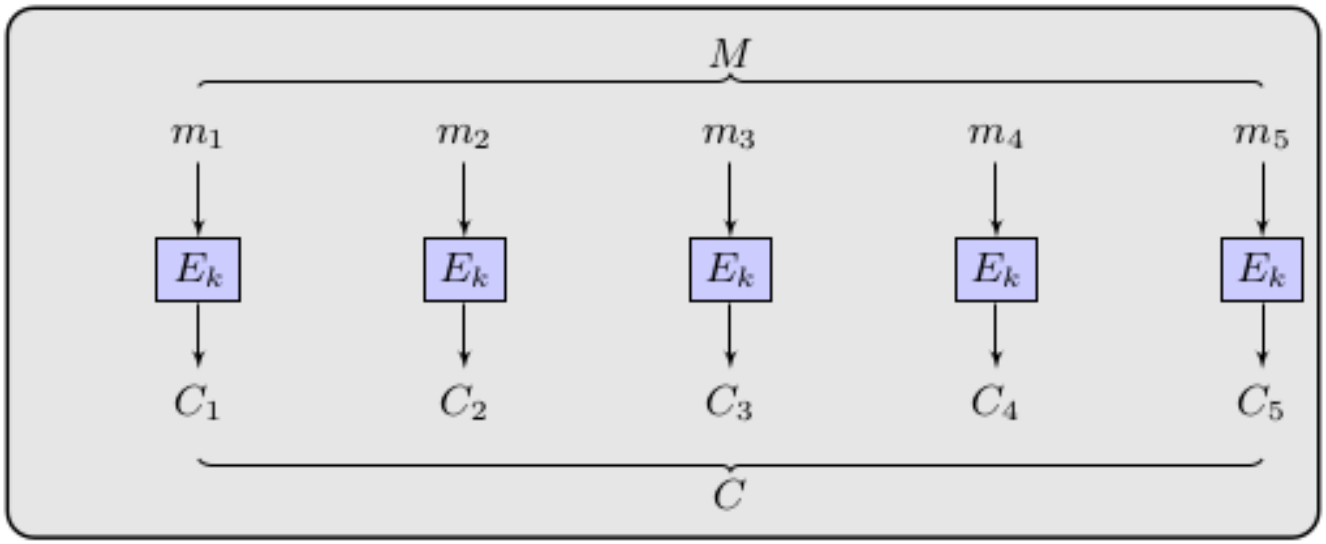


Figure: MOO Example: ECB

An Interesting Problem

We would like to be able to automatically generate different encryption schemes.

However, we don't want to have to prove or disprove CPA-Security for each of these schemes. We would like to have the security proof as part of the automated process.

Automated reasoning methods have been applies in other areas or cryptography, so we would like to also apply them here.

Automatic Synthesis

A third method is to start with some set of secure cryptographic primitives and then to automatically synthesis new cryptographic algoriothms out of those primitives. Symbolic methods can be used in several ways in this approach, in particular:

- They can be used to generate the new algorithms. This typically requires generating algorithms that follow some restrictions.
- They can be used to prove that the new algorithms are secure/insecure or that they meet some other security conditions.

Overview

We now focus on the following approach developed in Meadows [Meadows, 2020].

- First, a class of cryptosystems is defined and represented symbolically.
- Second, criteria on the symbolic representations are developed that guarantee security under the assumption that the cryptographic primitives used are indistinguishable from random.

Overview

In [Meadows, 2020] Meadows defines a symbolic condition on *MOO*-programs and their schedules, and shows that it is sound and complete with respect to IND$-CPA security.

Essentially, it captures the behavior of a symbolic adversary that is trying to use an encryption oracle to produce a sequence of ciphertext blocks that contains a subsequence that sums to zero with probability 1.

The adversary can only use information that it can compute itself or it has already seen. Thus it can be considered as attempting to solve a kind of constrained unification problem.

IND$-CPA security

IND$-CPA is short for "indistinguishability from random bits under an adaptive chosen-plaintext-and-IV attack"

- Security for encryption algorithms is generally defined in terms of some sort of indistinguishability property, e.g. an adversary should not be able to distinguish between the encryption of two different messages, or between the output of the cryptosystem and a string of random bits.
- IND$-CPA security, first studied by Rogaway in the context of modes of operation in [Rogaway, 2004], and later applied to the automatic generation and verification of modes by Malozemoff et al. in [Malozemoff et al., 2014]. Meadows considers IND$-CPA security [Rogaway, 2004].

$MOO_{\oplus}$ Programs

Symbolically we represent these systems as $MOO_{\oplus}$ - programs.

1. These can be thought of as a function from a sequence of plaintext blocks to a sequence of ciphertext blocks, together with a schedule for delivering the ciphertext blocks.
2. We can represent these as possible histories which describe the messages exchanged during a process in which an adversary interacts with the oracle to encrypt a message, where the encrypted message is a sequence of $MOO_{\oplus}$-terms.

Example: CBC

For example, Assume you have run three steps of the CBC program:

- The adversary starts a new sessions with the encryption oracle and send an initial nonce, IV.
- At each of the following steps:
  - The adversary sends the plaintext message P_i, represented by a variable x_i.
  - The Oracle returns $C_i = f(x_i \oplus C_{i-1})$
- The adversary ends the session.

We would then have the following symbolic history:

$$[IV, x_1, f(IV \oplus x_1), x_2, f(x_2 \oplus f(IV \oplus x_1))].$$

Connection to $MOO_{\oplus}$ Programs?

In checking IND$-CPA security, we can think of the adversary as attempting to break a mode of operation. The encryption oracle helps the adversary by responding to encryption requests. A mode of operation is IND$-CPA secure if the adversary has negligible advantage to distinguish between a sequence of blocks returned by mode of operation and a sequence of randomly generated blocks.

The adversary can distinguish between the two sequences if it can instantiate the plaintext in a way that causes two cipher-text blocks to be equal given our equational theory. This is called finding a collision in the literature. However, the adversary can't use any values when attempting to produce equal blocks. It can only instantiate a variable, x, with a term t if that term has been seen before x in the sequence of the *MOO*-program.

Example: ECB

Consider again the simple ECB example:

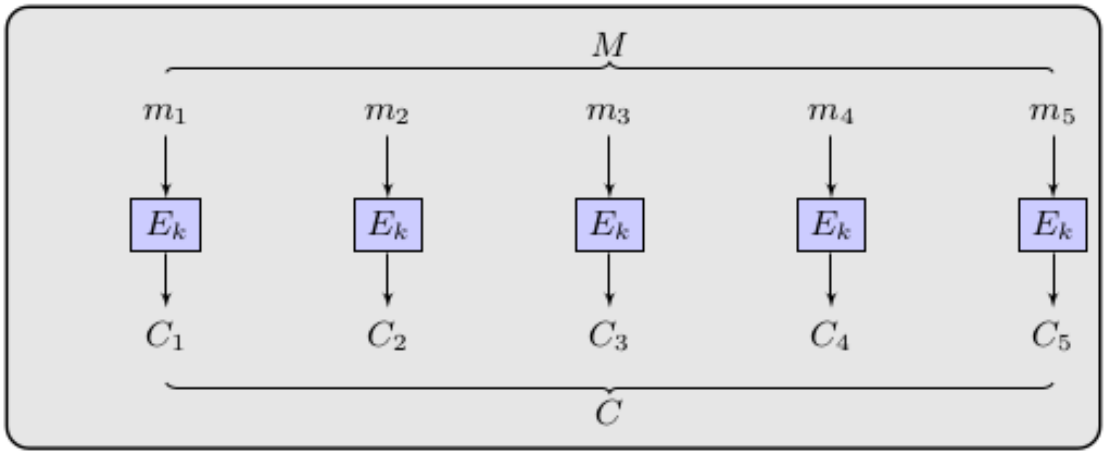


Figure: MOO Example: ECB

Notice if we have a symbolic history: $[x_1,\ f(x_1),\ x_2,\ f(x_2)]$, it's simple for the adversary to make two blocks equal simply by replacing the variables, x_1 and x_2, by the same term (or plaintext) we get the two cipher blocks $f(x_1) = f(x_2)$.

Example: ECB

Consider again the simple ECB example:

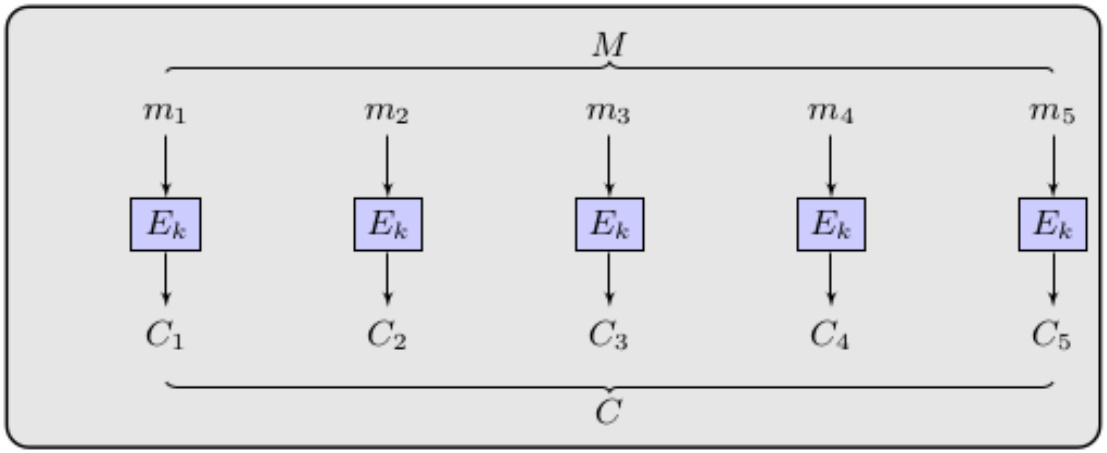


Figure: MOO Example: ECB

So, the adversary only has control over the plaintext, or the variables, however, it can still force a collision simply by finding a substitution on x_1 and x_2 that make the cipherblock terms equal.

The Symbolic Problems

The following problems are needed in either automating the above process or prove it's limitations.

- Results about decidability of IND$-CPA security.
  - Prove in general it's undecidable.
  - Show some conditions for which security is decidable.
- New Unification Problem needed for automating security.
- Invertibility, a security condition required in cryptographic schemes.

Decision problems

The first result is to show that for an arbitrary MOO-program, the security question is undecidable.

Decision problems: Encoding

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

- Let $PCP = \left(\frac{\alpha_0}{\beta_0}\right), \left(\frac{\alpha_1}{\beta_1}\right), \ldots, \left(\frac{\alpha_n}{\beta_n}\right)$.
- Let C_i be the i^{th} output of the cryptosystem.

The oracle will encode blocks as follows: For $i > 0$ let $C_i = E_{i_0}$ or, E_{i_2}, or $\ldots$, or E_{i_n} where,

- $E_{i_j} = [f(r_i \oplus C_{i,1}), f(r_i \oplus C_{i,2})],\ 0 \leq j \leq n$,
- $C_{i,1} = F(\alpha_j \bigoplus C_{i-1,1}),\ C_{i,2} = F(\beta_j \bigoplus C_{i-1,2})$,
- $C_{0,1} = F(\alpha_j \bigoplus 0),\ C_{0,2} = F(\beta_j \bigoplus 0)$.

Decision problems: Reduction

Here is how the program works:

- The adversary non-deterministically picks a possible solution to the PCP, $i_0, i_1, i_2, \ldots, i_k$.
- Each turn the adversary sends an index in the solution, starting with i_k and proceeding each turn until i_0 is reached.
- At each step the oracle encodes a pair of cipher blocks E_j and returns them to the adversary.
- After receiving each E_j, the adversary attempts to check if any two cipher blocks are equal.

The program stops if the adversary finds two equal pairs or the adversary stops the program.

Decision problems: Theorem

Assume M is an arbitrary non-deterministic *MOO*-program. The problem of determining if M, executing with a bounded number of sessions and unbounded session lengths, ever produces two equal cipher blocks is undecidable.

Example

Consider the following PCP:

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

A solution to this problem is $1, 3$.

Example

Let's trace a run of the program where the adversary guesses the solution $1, 3$.
-In the first step the adversary sends 3 to the oracle and receives the following cipher block in return. $C_0 = E_{0_3}$ where

$$E_{0_3} = [f(r_0 \oplus C_{0,1}), f(r_0 \oplus C_{0,2})]$$

$$C_{0,1} = F(\alpha_3 \bigoplus C_{0,1}) = f(a \oplus f(a \oplus f(a \oplus 0)))$$

$$C_{0,2} = F(\beta_3 \bigoplus C_{0,2}) = f(a \oplus f(a \oplus 0))$$

Example

At the second step the adversary sends a 1 to the oracle and receives the following in return. $C_1 = E_{1_1}$ where

$$E_{1_1} = [f(r_1 \oplus C_{1,1}), f(r_1, C_{1,2})]$$

$$C_{1,1} = F(\alpha_1 \bigoplus C_{0,1}) = f(b \oplus f(a \oplus C_{0,1}))$$

$$C_{1,2} = F(\beta_1 \bigoplus C_{0,2}) = f(b \oplus f(a \oplus f(a \oplus C_{0,2})))$$

Notice that now after step 2 the adversary has two cipher blocks, $C_{1,1}$ and $C_{1,2}$, which are equal.

$$C_{1,1} = f(b \oplus f(a \oplus f(a \oplus f(a \oplus f(a \oplus 0)))))$$

$$C_{1,2} = f(b \oplus f(a \oplus f(a \oplus f(a \oplus f(a \oplus 0)))))$$

Decision problems: Other Cases?

Other undecidability results using similar reduction:

- deterministic unbounded session length,
- both the deterministic and non-deterministic form of unbounded number of sessions with bounded session length.

A Decidable IND$-CPA Security Problem

While the question of IND$-CPA security is undecidable in general, there are identifiable conditions for which we can decide security.
Lin and Lynch were able to identify conditions that ensure decidability.

Checking Symbolic Security — Overview

- We introduce a property, called the *uniqueness property*, which implies symbolic security.
- We give an algorithm for checking the uniqueness property.

Motivating the *Uniqueness Property*

Consider a mode of operation M, where:
$C_i = f(t_i) \oplus x_i$
$C_0 = r$
Symbolic history:
$[r, x_1, f(t_1) \oplus x_1, x_2, f(t_2) \oplus x_2, x_3, f(t_3) \oplus x_3, \ldots]$.

Claim: Suppose that $f(t_0), f(t_1), \ldots$ are all unique (up to computable subsitutions), then M is symbolically secure.

Proof. (sketch) Consider any computable substitution σ. We know that $x_i\sigma$ can only be the xor of $f(t_j)$, where $j < i$.
$(C_1 \oplus \ldots \oplus C_n)\sigma = (f(t_1) \oplus \ldots \oplus f(t_n))\sigma \oplus (x_1 \oplus \ldots \oplus x_n)\sigma =$
$t \oplus f(t_n)\sigma$, for some t.

Uniqueness Property

Definition 2

Let M be a mode of operation. Consider any symbolic history H of M. M satisfies the *uniqueness property* if all of the f-rooted summands in H are unique (up to computable substitutions).

Theorem 7.1

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

Checking the Uniqueness Property — Overview

1. Assume that two f-rooted summands are the first (earliest) pair of unifiable terms in a symbolic history.
2. Try to derive a contradiction using some inference rules. Each rule is of the following form:

$$\frac{\{eq_1, eq_2, \ldots, eq_m\}}{\{eq'_1, eq'_2, \ldots, eq'_n\}}$$

   Semantics: If at least one of the equations in the premise hold, then at least one equation in the conclusion must hold.
3. If a contradiction can be derived, the mode of operation is secure. Otherwise, we do not know.

Inference Rules

$$\frac{\Gamma \cup \{f(t) \stackrel{?}{=} 0\}}{\Gamma} \; Elim_f$$

$$\frac{\Gamma \cup \{C_m \oplus C_n \stackrel{?}{=} 0\}}{\Gamma} \; Elim_C$$

where $m \neq n$.

$$\frac{\Gamma \cup \{C_m \oplus f(t) \stackrel{?}{=} 0\}}{\Gamma} \; Occurs\_check$$

where C_m is a subterm of t.

Inference Rules

$$\frac{\Gamma \cup \{f(t_1) \oplus \ldots \oplus f(t_n) \stackrel{?}{=} 0\}}{\Gamma \cup \{t_k \oplus t_1 \stackrel{?}{=} 0\} \cup \ldots \{t_k \oplus t_{k-1} \stackrel{?}{=} 0\} \cup \{t_k \oplus t_{k+1} \stackrel{?}{=} 0\} \cup \ldots \cup \{t_k \oplus t_n \stackrel{?}{=} 0\}} \; Pick_f$$

where k is chosen nondeterministically between 1 and n.

$$\frac{\Gamma \cup \{C_m \oplus f(t_1) \oplus \ldots \oplus f(t_n) \stackrel{?}{=} 0\}}{\Gamma \cup \{t'_u \stackrel{?}{=} t_1\} \cup \ldots \{t'_u \stackrel{?}{=} t_n\}} \; Pick_C$$

where (1) $f(t'_u)$ is an f-rooted summand of C_m. (2) The number of f-rooted summands is no more than n.

$$\frac{\Gamma \cup \{C_m \oplus f(t_1) \oplus \ldots \oplus f(t_n) \stackrel{?}{=} 0\}}{\Gamma} \; Pick_{fail}$$

where the number of f-rooted summands is greater than n.

An Example

Cipher Feedback (CFB) mode:
$C_i = f(C_{i-1}) \oplus x_i \ (i > 0)$
$C_0 = r$.

1. Assume that $f(C_{i-1})$ and $f(C_{j-1})$ $(i \neq j)$ are the first pair of unifiable f-rooted summands.
2. Apply inference rules.

$$\frac{\{f(C_{i-1}) \oplus f(C_{j-1}) \stackrel{?}{=} 0\}}{\{C_{i-1} \oplus C_{j-1} \stackrel{?}{=} 0\}} \ Pick_f$$

$$\frac{\{C_{i-1} \oplus C_{j-1} \stackrel{?}{=} 0\}}{\emptyset} \ Elim_C$$

3. A contradiction is derived. So Cipher Feedback (CFB) mode is symbolically secure.

Unification Problems

The next problem is a new form of the unification problem, which is needed to decide if a $MOO_{\oplus}$ program is secure.

In [Lin and Lynch, 2020] Lin and Lynch present an algorithm that can be used to answer the following question: given description of a ciphertext produced by a $MOO_{\oplus}$ in which logical variables stand for plaintext, is there any substitution to the variables computable by an adversary so that the resulting ciphertext contains a subsequence that sums to zero.

Invertibility

The final problem is invertibility, which is essentially the ability to decrypt.

- A natural requirement of any cryptographic algorithm is that it be *invertible*.
- This is typically not a problem for cryptographic methods developed by hand. However, for automatically generated systems, it needs to be checked.

Invertibility Result

Definition 3

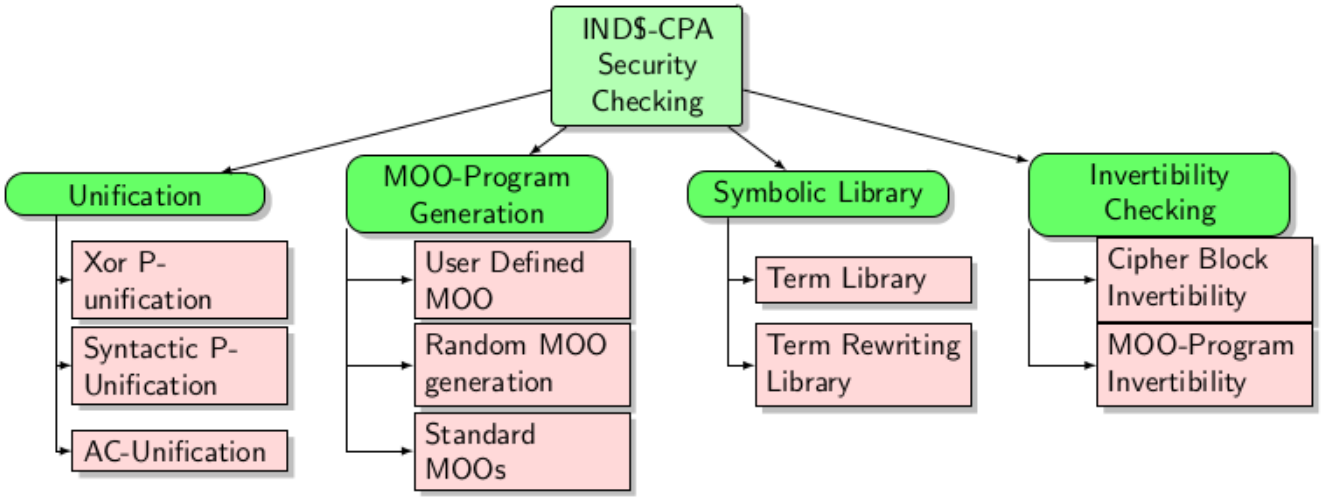
Consider recursive definitions which satisfy the following restrictions:

1. The base case, C_0, is a uniform random block is the only one used by the oracle in constructing the ciphertext.
2. C_i contains the i^{th} plaintext p_i.
3. p_i appears only once in C_i.

Lemma 7.1

Cryptosystems defined using Definition 3 are invertible, i.e., for all $i \geq 0$, p_i can be deduced from $\{C_0, \ldots, C_i\}$.

Tool Overview



Questions?

Thank you!

Lin, H. and Lynch, C. (2020).

Local xor unification: Definitions, algorithms and application to cryptography.
IACR Cryptol. ePrint Arch., 2020:929.

Malozemoff, A. J., Katz, J., and Green, M. D. (2014).

Automated analysis and synthesis of block-cipher modes of operation.
In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 140–152. IEEE.

Meadows, C. A. (2020).

Symbolic and computational reasoning about cryptographic modes of operation.
IACR Cryptol. ePrint Arch., 2020:794.

Rogaway, P. (2004).

Nonce-based symmetric encryption.
In *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, pages 348–359.