EasyCrypt - CodeCraftLab

- Mastering the Art of EasyCrypt Programming -

Ji, Yong-Hyeon

A document presented for the EasyCrypt

Department of Information Security, Cryptology, and Mathematics
College of Science and Technology
Kookmin University

July 16, 2024

Contents

1 Mathematical Background **4**

2 Cryptographic Background **5**
2.1 Attacker Type 5

3 Installing EasyCrypt **6**
3.1 Software Analysis 6
3.1.1 A Hard Limit 6
3.1.2 Trade-off 6
3.2 Basic Principle 6
3.2.1 Software Analysis based on Concrete Execution 6
3.2.2 Software Analysis based on Symbolic Execution 6
3.2.3 Software Analysis based on Abstract Execution 6

A Boolean Functions **7**

Symbols

In this paper, symbols are defined as follows.

$I \models P$ I satisfies P
$I \not\models P$ I does not satisfies P

Chapter 1

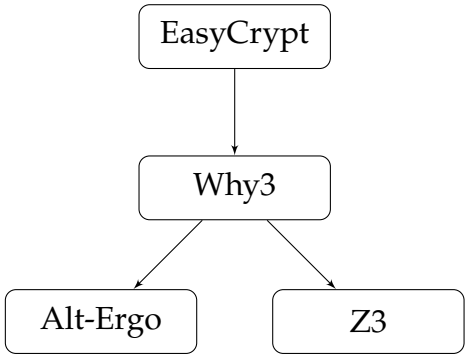
Mathematical Background

Chapter 2

Cryptographic Background

2.1 Attacker Type

- COA; Ciphertext-Only Attack
- KPA; Known-Plaintext Attack
- CPA; Chosen-Plaintext Attack
- CCA; Chosen-Ciphertext Attack
- CCA2; Adaptive Chosen-Ciphertext Attack



Integration of Why3, Alt-Ergo, and Z3 within EasyCrypt

- $\mathcal{K}$: key space
- $\mathcal{N}$: nonce space
- $\mathcal{P}$: plaintext space
- $\mathcal{C}$: ciphertext space

$$\texttt{used\_once}(n, \mathcal{N}) = n \in \mathcal{N}.$$

$$\begin{aligned} \texttt{used\_once} \quad : \quad \mathcal{N} \times \{\mathcal{N}\} &\longrightarrow \{0,1\} \\ (n, \mathcal{N}) &\longmapsto n \in \mathcal{N} \end{aligned}$$

$$\texttt{used\_once} : \mathcal{N} \to [\{\mathcal{N}\} \to \{0,1\}]$$

Chapter 3

Installing EasyCrypt

[1]

The official EasyCrypt installation instructions are available on the EasyCrypt GitHub. Below is a summary of these instructions that also emphasizes the connection with the Emacs text editor. EasyCrypt can be run from the shell (command line) in batch mode, to check individual .ec files. But when proofs are constructed interactively this is done within Emacs, with the generic interface Proof General mediating between Emacs and EasyCrypt, which is running as a sub-process of Emacs.

These instructions are current for:

- version 5.1.1 of the OCaml compiler;
- version 1.7.0 of why3;
- version 2.5.2 of alt-ergo.

(EasyCrypt is implemented in OCaml, why3 is the interface to SMT solvers used by EasyCrypt, and alt-ergo is one of SMT solvers you will need.)

3.1 Software Analysis

3.1.1 A Hard Limit

3.1.2 Trade-off

3.2 Basic Principle

3.2.1 Software Analysis based on Concrete Execution

3.2.2 Software Analysis based on Symbolic Execution

3.2.3 Software Analysis based on Abstract Execution

Appendix A

Boolean Functions

Bibliography

[1] Alley Stoughton. Easycrypt installation instructions, 2023. Accessed: 2024-07-12.