Paper Title

Diego Lupi, Pedro Nieto, and Huaira Gómez

FaMAF - Universidad Nacional de Córdoba, Córdoba, Argentina

Resumen Easycrypt[1] es una herramienta automatizada que soporta la construcción y verificación de pruebas de seguridad de sistemas criptográficos. Permite mejorar la confianza en sistemas criptográficos mediante la entrega de pruebas verificadas formalmente que resultan en sus metas propuestas. Provee una plataforma versátil que soporta pruebas automatizadas pero también permite al usuario realizar puebas complejas de manera interactiva entrelazando la verificación del programa con la formalización de las matemáticas, hecho fundamental al formalizar pruebas criptográficas. Con este paper nos proponemos mostrar las caracteristicas de esta herramienta y compararla con herramientas similares.

Keywords: Easycrypt · Game-based cryptographic proofs · Probabilistic.

1. Introducción

Desde siempre las pruebas criptograficas fueron propensas a errores, lo que naturalmente las puede llevar a ser erróneas. En particular en las pruebas de seguridad criptograficas la correctitud es critica para mejorar la confianza en el sistema criptografico. Actualmente se tiende a generar mas pruebas de seguridad de las que se pueden verificar y se omiten detalles finos desde un analisis formal que pueden tener grandes efectos en la practica. Teniendo en cuenta que los sistemas criptograficos en el mundo real pueden ser vulnerados, es necesario hacer las verificaciones sobre los pruebas de los sistemas criptograficos para evitar un desastre en el area de la seguridad.

Easycrypt es una herramienta automatizada que permite la construccion de pruebas de seguridad de sistemas criptograficos y su verificacion de manera interactiva usando la secuencialidad del codigo con un enfoque de game-based cryptographic proofs. Este enfoque consiste en la interaccion de un retador y un adversario, donde se especifica explicitamente la meta que adversario intenta alcanzar, como por ejemplo suponer de manera correcta una porcion de informacion oculta. En Easycrypt los juegos criptograficos se modelan como modulos, que consisten en procedimientos escritos en lenguaje propio de la herramienta. Por otra parte los adversarios se modelan como modulos abstractos, modulos cuyo codigo es desconocido y puede cuantificarse.

Posteriormente se sumo al desarrollo la École Polytechnique (Escuela Politecnica). IMDEA software institute es un instituto para el estudio avanzado de

tecnologias para el desarrollo de software asentado en Madrid, España. Inria es un centro de investigación francés especializado en Ciencias de la Computación, teoría de control y matemáticas aplicadas. Por ultimo, la École Polytechnique es una gran escuela de ingenieros francesa bajo la tutela del Ministerio de Defensa.

El primer prototipo de EasyCrypt lanzado en 2009 fue desarrollado por IMDEA Software Institute, e Inria. Constaba de una interfaz de linea de comando y funcionalidades muy acotadas. Posteriormente se sumo al desarrollo la École Polytechnique (Escuela Politecnica). IMDEA software institute es un instituto para el estudio avanzado de tecnologias para el desarrollo de software asentado en Madrid, España. Inria es un centro de investigación francés especializado en Ciencias de la Computación, teoría de control y matemáticas aplicadas. Por ultimo, la École Polytechnique es una gran escuela de ingenieros francesa bajo la tutela del Ministerio de Defensa. En el año 2012 se le hizo una reimplementacion completa al prototipo con el objetivo de superar varias de las limitaciones que este revelo. Actualmente se encuentra en la version 1.0 que fue liberada el 10 Octubre de 2017. En esta version los desarrolladores permitieron que EasyCrypt pueda ejecutar scripts interactivamente en Proof General[2], dandole a la herramienta una interfaz grafica interactiva en la que el usuario puede simular paso a paso la verificacion de su especificacion, otorgando la posibilidad al usuario de elegir el enfoque por el cual quiere verificar la misma. Por otro lado para proveer las bases requeridas para llevar a cabo algunos razonamientos criptograficos estandares se implementaron cuatro logicas, lo que permite realizar pruebas mas complejas, que en versiones anteriores no eran verificables.

2. Caracteristicas de EasyCrypt

EasyCrypt esta diseñada para verificar pruebas criptograficas de manera estructurada. La herramienta puede ayudar a corregir errores y obtener la seguridad probable de sistemas criptograficos. Estos sistemas practican la comunicacion segura, ante la presencia de terceros, en los ambitos de comercio electronico, crypto-monedas, claves de computadoras, tarjetas de pagos con chips y comunicaciones militares.

La herramienta permite codificar y verificar game-based proofs, pero tiene distintos lenguajes para distintas tareas. El principal lenguaje de especificacion de EasyCrypt es el lenguaje de expresiones, en el cual se definen los tipos junto con los operadores que se pueden ser aplicados. Este lenguaje soporta el polimorfismo parametrico. Por otra parte, los lenguajes de expresiones no son adecuados para definir juegos y otras estructuras de datos como esquemas criptograficos y oraculos, debido a la naturaleza dependiente del estado previo de los algoritmos secuenciales. Por eso EasyCrypt usa un lenguaje diferente, llamado pWhile[3] (probabilistic while) para definirlos:

Cuadro 1. Lenguaje pWhile

$\mathcal{C}$ ::= skip	nop
$\mid \mathcal{V} \leftarrow \mathcal{E}$	assignment
$\mid \mathcal{V} \xleftarrow{\$} \mathcal{D}\mathcal{E}$	random sampling
$\mid$ if $\mathcal{E}$ then $\mathcal{C}$ else $\mathcal{C}$	conditional
$\mid$ while $\mathcal{E}$ do $\mathcal{C}$	while loop
$\mid \mathcal{V} \leftarrow \mathcal{P}(\mathcal{E},...,\mathcal{E})$	procedure call
$\mid \mathcal{C}; \mathcal{C}$	sequence

La herramienta se restringe a la etapa de verificacion del desarrollo de software. En el trabajo Mind the Gap: Modular Machine-checked Proofs of One-Round Key Exchange Protocols[4] se desarrolla una nueva prueba de seguridad generia para protocolos intercambio de llaves, y se lo instancia para obtener pruebas de seguridad de protocolos conocidos respecto a distintos modelos de adversarios usando EasyCrypt.

2.1. Aspectos tecnicos

Referencias

1. Gilles Barthe, Juan Manuel Crespo, Benjamin Gregoire, Cesar Kunz, Santiago Zanella Beguelin. Computer-Aided Cryptographic Proofs. Third International Conference, ITP, 2012.
2. Proof-General Website: (2016) https://proofgeneral.github.io.
3. G. Barthe, B. Grégoire, and S. Zanella Béguelin, "Probabilistic relational hoare logics for computer-aided security proofs," in Mathematics of Program Construction (J. Gibbons and P. Nogueira, eds.), vol. 7342 of Lecture Notes in Computer Science, pp. 1–6, Springer Berlin Heidelberg, 2012.
4. Gilles Barthe, Juan Manuel Crespo, Yassine Lakhnech, Benedikt Schmidt. Mind the Gap: Modular Machine-checked Proofs of One-Round Key Exchange Protocols. 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015.