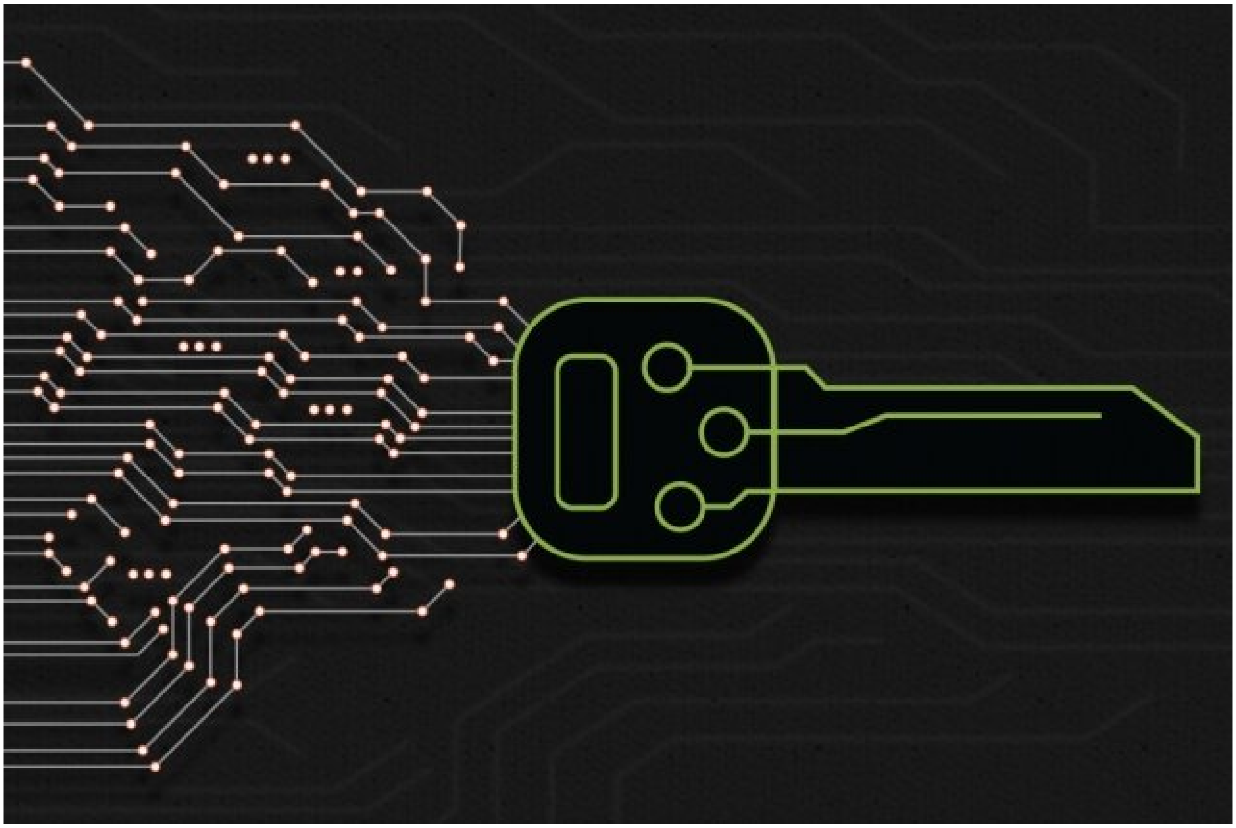
TechXplore

Automated cryptocode generator is helping secure the web

June 18 2019, by Rob Matheson



“Fiat Cryptography,” a system developed by MIT researchers, automatically generates — and simultaneously verifies — cryptographic algorithms optimized across all hardware platforms. Algorithms generated by the system are already behind most of the secure links opened in Google Chrome. Credit: Chelsea Turner, MIT

Nearly every time you open up a secure Google Chrome browser, a new MIT-developed cryptographic system is helping better protect your data.

In a paper presented at the recent IEEE Symposium on Security and Privacy, MIT researchers detail a system that, for the first time, automatically generates optimized cryptography code that's usually written by hand. Deployed in early 2018, the system is now being widely used by Google and other tech firms.

The paper now demonstrates for other researchers in the field how automated methods can be implemented to prevent human-made errors in generating cryptocode, and how key adjustments to components of the system can help achieve higher performance.

To secure online communications, cryptographic protocols run complex mathematical algorithms that do some complex arithmetic on large numbers. Behind the scenes, however, a small group of experts write and rewrite those algorithms by hand. For each algorithm, they must weigh various mathematical techniques and chip architectures to optimize for performance. When the underlying math or architecture changes, they essentially start over from scratch. Apart from being labor-intensive, this manual process can produce nonoptimal algorithms and often introduces bugs that are later caught and fixed.

Researchers from the Computer Science and Artificial Intelligence Laboratory (CSAIL) instead designed "Fiat Cryptography," a system that automatically generates—and simultaneously verifies—optimized cryptographic algorithms for all hardware platforms. In tests, the researchers found their system can generate algorithms that match performance of the best handwritten code, but much faster.

The researchers' automatically generated code has populated Google's BoringSSL, an open-source cryptographic library. Google Chrome, Android apps, and other programs use BoringSSL to generate the various keys and certificates used to encrypt and decrypt data. According to the researchers, about 90 percent of secure Chrome communications currently run their code.

"Cryptography is implemented by doing arithmetic on large numbers. [Fiat Cryptography] makes it more straightforward to implement the mathematical algorithms … because we automate the construction of the code and provide proofs that the code is correct," says paper co-author Adam Chlipala, a CSAIL researcher and associate professor of electrical engineering and computer science and head of the Programming Languages and Verification group. "It's basically like taking a process that ran in human brains and understanding it well enough to write code that mimics that process."

Jonathan Protzenko of Microsoft Research, a cryptography expert who was not involved in this research, sees the work as representing a shift in industry thinking.

"Fiat Cryptography being used in BoringSSL benefits the whole [cryptographic] community," he says. "[It's] a sign that the times are changing and that large software projects are realizing that insecure cryptography is a liability, [and shows] that verified software is mature enough to enter the mainstream. It is my hope that more and more established software projects will make the switch to verified cryptography. Perhaps within the next few years, verified software will become usable not just for cryptographic algorithms, but also for other application domains."

Joining Chlipala on the paper are: first author Andres Erbsen and co-authors Jade Philipoom and Jason Gross, who are all CSAIL graduate students; as well as Robert Sloan MEng '17.

Splitting the bits

Cryptography protocols use mathematical algorithms to generate public and private keys, which are basically a long string of bits. Algorithms use these keys to provide secure communication channels between a browser and a server. One of the most popular efficient and secure families of cryptographic algorithms is called elliptical curve cryptography (ECC). Basically, it generates keys of various sizes for users by choosing numerical points at random along a numbered curved line on a graph.

Most chips can't store such large numbers in one place, so they briefly split them into smaller digits that are stored on units called registers. But the number of registers and the amount of storage they provide varies from one chip to another. "You have to split the bits across a bunch of different places, but it turns out that how you split the bits has different performance consequences," Chlipala says.

Traditionally, experts writing ECC algorithms manually implement those bit-splitting decisions in their code. In their work, the MIT researchers leveraged those human decisions to automatically generate a library of optimized ECC algorithms for any hardware.

Their researchers first explored existing implementations of handwritten ECC algorithms, in the C programming and assembly languages, and transferred those techniques into their code library. This generates a list of best-performing algorithms for each architecture. Then, it uses a compiler—a program that converts programming languages into code computers understand—that has been proven correct with a proofing tool, called Coq. Basically, all code produced by that compiler will always be mathematically verified. It then simulates each algorithm and selects the best-performing one for each chip architecture.

Next, the researchers are working on ways to make their compiler run even faster in searching for optimized algorithms.

Optimized compiling

There's one additional innovation that ensures the system quickly selects the best bit-splitting implementations. The researchers equipped their Coq-based compiler with an optimization technique, called "partial evaluation," which basically precomputes certain variables to speed things up during computation.

In the researchers' system, it precomputes all the bit-splitting methods. When matching them to a given chip architecture, it immediately discards all algorithms that just won't work for that architecture. This dramatically reduces the time it takes to search the library. After the system zeroes in on the optimal algorithm, it finalizes the code compiling.

From that, the researchers then amassed a library of best ways to split ECC algorithms for a variety of chip architectures. It's now implemented in BoringSSL, so users are mostly drawing from the researchers' code. The library can be automatically updated similarly for new architectures and new types of math.

"We've essentially written a library that, once and for all, is correct for every way you can possibly split numbers," Chlipala says. "You can automatically explore the space of possible representations of the large numbers, compile each representation to measure the performance, and take whichever one runs fastest for a given scenario."

More information: Simple High-Level Code For Cryptographic Arithmetic – With Proofs, Without Compromises. adam.chlipala.net/papers/FiatC … 9/FiatCryptoSP19.pdf

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: Automated cryptocode generator is helping secure the web (2019, June 18) retrieved 6 May 2024 from https://techxplore.com/news/2019-06-automated-cryptocode-web.html