Report on COINS Summer School on Secure Implementation of Cryptographic Software - 2017

by PhD Candidate Andrii Shalaginov
andrii.shalaginov@ntnu.no
NTNU i Gjøvik, Norway

September 8, 2017

Abstract

This report reflects content of a COINS summer school that happened on 27th of August - 3rd of September 2017 with a special focus on Secure Implementation of Cryptographic Software. There were given theoretical lectures as well as practical exercises and tutorials with hands-on on real hardware development broads. Below the summery of every day is given together with a defined questions related to the presentations.

Keywords: Crypto, Embedded Devices, Software Development, Side-channel Attacks.

Day 1. Paris Kitsos from *Digital IC dEsign and Systems Laboratory*[1] gave an extensive presentation on Hardware Trjoan Horse (HTH). These are not the malicious software targeting hardware, but rather a malicious modification of the circuit. Such Trojan Horses are introduced on the design level and can be hard to detect. They trigger some kind of activity and are usually used to bypass some security functionality. HTH have two main components: Trigger and Payload. The payload is being executed once the trigger has been flipped under some conditions defined by sensors or internal state. Also during the lecture we learned about VLSI chips, more general FPGA and more specific ASIC integrated circuits. Moreover, Paris also introduced a concept of a "Golden Chip" that is a benign component designed and implemented in a physical circuit and used to verify design and implementation of similar chip from other manufacturer. Generally, there is no one-fits-all approach in detecting the HTH.

During the practical exercises students Vivado HLX 2016 together with Digilent Basys 3 FPGA development boards to perform practical tasks such that using Ring Oscillator to count the number of cycles in order find if there is an actual HTH present.

Day 2. Lejla Batina from Digital Security Group at Nijmegen[2] gave a good overview of the possible side-channel attacks and ways to perform power analysis. The students learned about a secret program called TEMPEST[3] that was recently declassified by NSA. The main idea behind this program that was under secret label for many decades is to use side-channel attacks to retrieve classified information. The two general ways of performing this attack is to use either Simple Power Analysis (SPA) or Differential Power Analysis (DPA). The SPA is done through simple measurement of Electromagnetic field around the devices that is being targeted. On the other hand, the DPA is done through comparison in output signals switching. This also might include Static and Dynamic power consumption. On the other hand, Pearson correlation might be useful for key guesses over large number of collected measurements.

[1] http://www.diceslab.cied.teiwest.gr/index.php/en/
[2] http://www.ru.nl/ds/
[3] https://rdist.root.org/2008/04/30/history-of-tempest-and-side-channel-attacks/

The practical exercises included doing full range of the side-channel attacks using power consumption prediction. It was clear from the exercise that such attacks does not have very high complexity and under some circumstances can be performed even in time and resources constrained environment. The counter-measures include Faraday Cage or one need to destroy a link between intermediate values and power consumption such that by using NOP operations.

Day 3. Peter Schwabe from Radboud University[4] gave a very good overview of how the implementation of cryptographic algorithms can be optimized for a use on embedded devices. This sessions was very practical. At the beginning, Peter gave a talk on different aspects of cryptographic algorithms application on embedded devices together with an overview of the recent methods such that ChaCha20 (stream cipher that was published in 2008 and is used in TLS now) and Poly1305 (cryptographic message authentication code also used in TLS)[5]. Following this, the participants received STM32 development micro-controller boards to experiments with the embedded implementation of the aforementioned algorithms on such board. The students worked in pairs to map the C code of the cryptographic algorithm to Assembly code with further optimization using machine architecture aspects. Meanwhile, it was a good chance to revise the knowledge of machine instructions and assembly principles.

Day 5. Justin Cappos from New York University[6] presented his lecture on "Securing software development for computationally weak devices". In particular, the students were excited to hear about the following type of attacks that also can be used in updates on Linux distros: slow retrieval (very slow or limited bandwidth), endless data (the server does not stop sending unlimited amount of data causing depletion of either disk or RAM space), ZIP bomb (42.zip with a size of 42 Kilobytes that contains five layers of nested zip files in a set of 16 resulting in 4.5 Petabytes of uncompressed data), Duplicate Files in Zip (Google APK may contain several files with the same name with a hope that the malicious will be retrieved), Arbitrary Modify Updates, Freeze Attack (the attacker prevents the whole set of updates from updating), Mix and Match Attack (supplies unique combination of packages that never existed before in the repository), Depends on Everything (multiple nested dependencies in a package). Justin also described a Stork secure package manager that he developed and maintained for a long time.

Day 6. On the last day, Mark D. Ryan from the University of Birmingham[7] gave a nice talk on Intel SGX[8] technology and all possible aspects of its application. We learned about the enclaves both in code and memory domains. Intel SGX is present on Intel Skylake CPU onwards. This is considered to be a trusted platform module that can be used by software developers to protect any kind of data during the execution. However, this requires also explicit permission from Intel to run the defined software in an enclave. This is similar to ARM TrustZone that had been added to Cortex processed to allow to run a secure operating system.

Later on Alexandra Weber from TU Darmstadt[9] presented her research on secure refinement of the cryptographic algorithms, reliable side-channel security in particular. One of the most famous examples is a "Square and Multiple" case, where Alexandra presented how the side-channel attacked can be carried out on a straight-forward C implementation. This is basically a timing attack used to measure the time it requires to execute the algorithm and measure how the time differs under variety of conditions.

[4] https://cryptojedi.org/peter/index.shtml
[5] https://tools.ietf.org/html/rfc7539
[6] https://ssl.engineering.nyu.edu/personalpages/jcappos/
[7] https://www.cs.bham.ac.uk/~mdr/
[8] https://software.intel.com/en-us/sgx
[9] http://www.mais.informatik.tu-darmstadt.de/Alexandra_Weber.html