ECE/CS 5580 Cryptographic Engineering
Spring 2018
Project Specification

1 Objectives

The objective of the course project is to implement an AES cryptosystem, from specification down to implementation, and demonstrate an attack and a countermeasure to that attack. The project has four phases which each emphasize a specific activity during the implementation/attack/defense of the selected cryptosystem.

1. **Executable Specification and Implementation**. You will build a high-level model of AES as a golden reference model. You will use this model to study how the cryptographic algorithm works, and to generate test-vectors of the selected implementation. You will then map the specification onto a selected software platform, making sure that the implementation is functionally equivalent to the high-level executable specification. This model helps you to understand the features of the target platform, and the potential challenges for attack/countermeasures.

2. **Proof-of-concept Attack**. From the literature you will select a state-of-the-art, power-based side-channel attack and prove its efficacy using publicly available power traces. You will need to explain the innovation of the attack over the basic attack side-channel attack it is based on (CPA, DPA, or MIA).

3. **Execution of Attack**. You will investigate the transferability of the chosen attack against your own AES implementation by collecting power traces from it during encryption. To ease collection of traces, you may insert code/use additional hardware that triggers the data collection instrument immediately preceding and after the encryption operation. These traces will be made available to the wider community so that researchers can study the efficacy of attacks across platforms.

4. **Enactment/validation of a Countermeasure**. From the literature you will select, and implement on your platform, a countermeasure meant to impede or prevent power-based side-channel attacks. You

will demonstrate the efficacy or inefficacy of the countermeasure by collecting another set of power traces from your implementation, with the countermeasure in place, and attempt the same attack as before. Traces will again be made available to the wider community.

Each phase is completed by writing a report, and submitting a typeset version of it on the course website. The report will be graded and returned as a guidance for the next phase. The due dates for each report are 8 February, 1 March, 10 April, and 8 May, respectively. There will be no extensions for late reports.

To ensure regular progress, you will need to document the project activities in a blog. There needs to be at least one entry per week, per project in each blog. Furthermore, everyone is encouraged to read other blogs and comment on it.

A key difference between contributions in the written report, and contributions online in the blogosphere, is that reports are graded per group, while blog entries are individually graded. The reports and blogs, together, count for an important portion of the course grade (70 %).

2 Timeline

- *No later than 30 January:* Form a group of no more than 3 students, announce to the TA (ymo6@vt.edu) by email. We will confirm the groups by 31 January.

- *No later than 1 February:* Select target platform and announce to the TA (farhady@vt.edu) by email. We will confirm the platform by 8 February.

- 1 February: Blog Setup, Blog Post 1 posted. Topic for discussion: review of five different embedded systems to implement AES on/perform side-channel attack against and criteria for selection. Keep in mind that the oscilloscopes you'll have access to have a bandwidth of 250 MHz and sample rate of 2.5 GS/s so the max (unsync'd) clock rate of the device should be 125 MHz.

3 Group Formation, Topic Selection

Students are required to form a group of no more than 3 students, *no later than 30 January*. Email the student names to the TA (ymo6@vt.ed).

After you have formed the group and identified it to the instructor, you will be assigned a blog website and a website administrator account. You can then proceed setting up the blog.

Platform selections will have to be unique (ie. no groups will be allowed to work on the same target platform, simultaneously). When groups will select the same combination, the assignment will be made first-come, first-served. If too few platforms are readily available, we will differentiate groups based on the measurement point (e.g., high- vs. low-side measurements).

Note that the selected platforms must be *software*. Hardware target platforms (ASIC, FPGA) will only be allowed provided that a compelling reason for their study (over software platforms) can be provided. The following architectures may be used as a starting point (a platform is a specific implementation of an architecture).

- (32-bit) ARM Cortex-M
- (16-bit) MSP-430
- (8-bit) Atmel AVR

4 Code Development and High-Level Specification

The end objective in this project is the detailed design of a secure cryptosystem. This requires programming and coding at multiple levels of abstraction and possibly using multiple programming languages.

Code development needs to follow common code quality standards (testing; comments; build automation; etc). Code development needs to make use of version control. Every group is encouraged to create a repository on github to facilitate sharing of code among all group members.

The initial implementation of the cryptosystem needs to be done in a high-level language, to be selected from one of the following three options. It's important to have a flexible high-level specification mechanism that helps you in the creation of test vectors.

- SAGE, a computer algebra system; `http://www.sagemath.org`
- Cryptol, a functional language for cryptography; `http://www.cryptol.net/`
- Python, a high level scripting language with rich library support; `https://www.python.org/`

The detailed design of the cipher will need to be done on C, C++, or assembly code, depending on the chosen platform, and depending on the chosen attack. The attack itself can be scripted in Python or Matlab.

5 Attacks and Countermeasures

Extensions to basic AES side-channel attacks and countermeasures may be found from

- The CHES Workshop; `https://ches.iacr.org`
- IEEE Transactions on Information Forensics and Security; `http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=10206`
- DPA Contest; `http://www.dpacontest.org/home/`
- Google Scholar; `https://scholar.google.com`

You will be provided access to a laboratory for collecting the power traces. Available equipment and parts include: power supplies, oscilloscopes, amplifiers, filters, single-ended/differential probes, and cables/converters. We have three oscilloscopes available so some amount of coordination and planning will be necessary. It is therefore imperative that your collection routines be as fast and efficient as possible.

6 Balancing Individual Effort and Group Effort

One part of the grade of this project (project reports) is given to the group; another part (blogging and comments) is given to the individual. How do you balance these two?

- There are 4 reports and 12 blog posts due over the entire semester. Hence, your group will need to plan and partition the effort fairly. Make a planning and partition the effort among team members. You can blog, or comment on blogs, multiple times. You can participate in part of each report (recommended), or write an entire report by yourself.

- Some of your team members may be strong algorithm people, others may be better implementers. Partition the effort, making optimal use of everybody's skill. Collaborate, don't compete.

- Choose a team leader and trust this individual with being a manager for your project. A team leader is *not* the most visible individual of the team. A team leader is an individual who cares about gluing the project pieces together, and who cares about collaboration. Being a team leader does not bring extra credit. Producing an outstanding result, on the other hand, *will* bring extra credit.

- Do not procrastinate. Developing code, be it in Python, C or assembly, takes time. Debugging takes time. Measurements take time. Make a plan for each part of the project, with the report as the outcome of that phase. If your report needs to demonstrate tangible results (testvectors, code samples, cycle counts, ...), then make sure that you have the results *before* writing the report.

7 Grading criteria for Blogs

- We encourage you to document your developments and your research process for your project on a blog. This can include anything that you deem appropriate for the project: experiments done; papers read and analysed; new ideas for measurements; observations and items learned from the class lectures; and so forth. The blog is a living record of your efforts in this project.

- We expect every team member to contribute to your team blog on a regular basis. We expect every team member to contribute to comments on any blog of the 5580 course. Of course, you do not have to feel compelled to make a comment on every group and every project.

However, you should follow up on projects that are related to yours (for example, because both of them relate to a differential power attack).

- We will regularly comment on your blog entries as well and we will encourage you to add more material or elaborate where we think more detail is needed.

- A good example of a blog with high-quality articles is the Bristol Crypto Blog (`http://bristolcrypto.blogspot.com/`). Of course, this blog does not discuss a project. It is (mostly) written, though, by graduate students at Bristol University in UK.

- Your blog grade is give on the posts you make, as well as on the comments you add to other blogs. We will evaluate your engagement in this class by checking how often you read and interact with the blog - by posting articles or by posting comments. We will try to give you a 'blogging' grade and a 'blogging comment' grade about three times over the course of the semester. This will enable you to adapt or improve your style based on the grade you receive.

8 Ten tips for a preparing a solid project report

In my experience, a rejected conference paper is often the result of a mismatch between what the author wants to say and what the reviewer expects to read (sometimes, but not always, it's the result of poor scientific quality). Being clear and unambiguous is of crucial importance. Here are some tips to create a strong write-up.

- **Use a clear problem statement.** An amazing amount of conference paper proza is being written without a clear problem statement. We're engineers, and problem solving is our bread and butter. If you can't clearly explain what you're trying to address, don't bother explaining a solution for it! To explain a problem, relate to technical challenges and roadblocks, and (in the case of this course) security flaws and risks. A problem statement works as an attention-getter. It could correspond to what you would tell to someone (a sponsor, for example) to capture his or her interest without disclosing your solution.

- **Minimize context.** When being immersed for a long time in a specific topic, you tend to become a specialist, and you start to use the lingo and abstractions that are most common to the area you work in. This is called 'context': the body of knowledge that you have acquired to work on this specific topic. Someone who is new to your area does not have this context, for various reasons. The challenge then is to explain your problem and solution while minimizing the amount of context (or background knowledge) needed. That is not easy. Too much context makes the paper boring of too complicated. However, insufficient context may make it incomprehensible. You have to filter out the essential elements of your problem and solution, and capture those completely and clearly in the write-up.

- **Define clear quality metrics.** Being able to do a good job becomes easier if you explain what it means to do a good job. Quality metrics are preferably quantifiable (eg. gate count, power dissipation, cycles per byte, ..). In addition, it also helps to define absolute bounds ('must be smaller than 1000 gates, and consume less than 2 mJ energy').

- **Write for all ages.** Use a simple writing style. Use short, active sentences.

- **Related work.** No research effort happens in complete isolation. Make sure to put adequate references to any related work you rely on. Include properly formatted references at the end of the paper. Make sure that the references are relevant. When you write a paper on AES, it's quite easy to make a long list of papers that mention a thing or two about AES, but this does not mean that all these papers will be relevant for your particular research. Many conference papers make the mistake of including a 'related work' section that collects a number of arbitrary references to other people's work. Don't fall into this trap: make sure that you know exactly why a reference is included in your paper. If you don't have a reason to reference, don't!

- **Graphics are good for engineers.** You can explain complex ideas with a single figure. Use block diagrams to your advantage, and introduce them early on in the paper to document how everything fits together. Some domains, like hardware design, lend themselves very well to the use of graphics. Other domains, like software design, are

harder to support with graphics. When using charts, make a special effort to optimize the quality of your chart. Microsoft Excel is a clumsy tool when it comes to graphs. Check these ten-worst-graphs as examples of how NOT to use graphs (`http://www.biostat.wisc.edu/~kbroman/topten_worstgraphs/`).

- **Analyze your experimental data.** If your project includes collecting experimental data, make sure to analyze it. Just putting the numbers and leaving the conclusion to the imagination of the reader is not helpful. The analysis of your experimental data will lead to better understanding and appreciation of your results.

- **Don't forget the conclusions.** Conclusions are not the summary of your paper; conclusions are the logical consequence of the experiments and analysis you describe in the paper. A conclusion may or may not include suggestions for future work, but it should always confirm the claims made in the abstract.