TTM4205: Technical Essay Fall 2025

Tjerand Silde and Caroline Sandsbråten

{tjerand.silde, caroline.sandsbraten}@ntnu.no

Assignment

This is one out of three assignments in the course TTM4205 Secure Cryptographic Implementations (ttm4205.iik.ntnu.no) during fall semester of 2025.

This assignment is to write a *technical essay* and give a *presentation* about a scientific topic related to the content given in the course description: either a topic not covered by the lectures or a topic from the lectures more in-depth. It will be joint work in groups of two or three, and the essay should be *roughly* 8 *to* 10 *pages* long, in addition to references. The topic, scope, and group must be *approved* by the staff. General information:

- Cheating at NTNU: i.ntnu.no/wiki/-/wiki/English/Cheating+on+exam
- Writing tips at NTNU: i.ntnu.no/en/academic-writing/write-academically
- Bibliographic references to cryptology papers: publish.iacr.org/cryptobib
- CS paper: froihofer.net/students/how-to-write-a-computer-science-paper
- NTNU style guide: ntnu.edu/english-matters/ntnu-english-style-guide

All essays and presentation slides must be written in LaTeX, and we provide mandatory templates to be used at overleaf.com/read/nhcnrbnwzmcw (essay) and overleaf.com/read/zjqxggmjnzqp (presentation).

This assignment counts for at most 40 points, based on the following criteria: scientific correctness, quality of writing, the structure of the essay, presentation (figures/tables), referencing, relevant and consistent background material, clear and detailed main section(s), and justification of conclusions.

The topic must be approved by October 31st, but we recommend starting earlier. If you want the staff to provide feedback on your essay, you can submit a draft by November 21st and get a response by December 5th. Oral presentations will be in mid to late November, to be scheduled later.

Submission deadline: December 19th at 23:59 in Ovsys2.

Suggested Topics

We suggest some relevant topics for the technical essay below, but you can also suggest your own. In the former case, you need to detail the scope of the essay yourself, and we allow for at most two groups working on the same high-level topic. In the latter case, you are expected to provide a (preliminary) title and scope, in addition to at least two references.

1 Cryptographic Fuzzing and Static Analysis

It is hard to verify if a given piece of cryptographic code is securely implemented or not. Vulnerabilities include side-channel leakage, lacking API checks, and correctness errors. One possible solution to detect these mistakes is cryptographic fuzzing [Som16]; see also github.com/kudelskisecurity/cdf. Furthermore, static analysis can be used to discover the wrong usage of randomness or cryptographic algorithms [LJL+22].

2 Formally Verified Cryptographic Code

While cryptographic fuzzing and static analysis are excellent approaches to finding vulnerabilities, they are reactive solutions that require much work after the code is written. A more proactive approach is only to allow correct and secure code to be written in the first place by disallowing insecure algorithms, automatically generating code [EPG+20], and using languages that do not compile if certain functions or operators are used [ZBPB17].

The talk by Filippo Valsorda on the design of the Go Crypto Library at infoq.com/presentations/go-crypto-library and the blog post by Microsoft on Project Everest at microsoft.com/en-us/research/blog/project-everest-reaching-greater-heights-in-internet-communication-security discuss this.

3 Vulnerabilities in Threshold Signatures

Lindell published a threshold signature scheme [Lin21] based on the Elliptic Curve Digital Signature Algorithm (ECDSA). This scheme was later implemented and used in practice, and, while the construction was theoretically secure, the implementations contained bugs that allowed an attacker to extract the secret key [AS20]. See also the report available at fireblocks.com/blog/lindell17-abort-vulnerability-technical-report. There are some more recent attacks on threshold ECDSA by Makriyannis, Yomtov and Galansky [MYG23] on similar schemes, presented at a NIST workshop: csrc.nist.gov/presentations/2023/mpts2023-day2-talk-attack-threshold-ecdsa-wallets.

4 Degenerate Edwards Curve Attacks

We get into trouble if we do not verify that points $P = (x, y)$ are on the (Weierstrass) elliptic curve $E(\mathbb{F}_p) : y^2 = x^3 + a \cdot x + b$ [ABM+02] (see; Weekly Problems). Furthermore, the addition formulas are not complete, which means that the way we compute the addition of two points P and Q on $E(\mathbb{F}_p)$ depends on the input. This makes implementation more complicated to get correct and enforces complex side-channel protection measurements since the difference in addition method may leak secret key data.

Edwards curves $Ed(\mathbb{F}_p) : y^2 - x^2 = 1 + d \cdot x^2 \cdot y^2$ (simplified) were introduced to solve these issues, leading to the more efficient and secure signature scheme EdDSA [BDL+11], which was later standardized by the Internet Research Task Force (IRTF) at datatracker.ietf.org/doc/html/rfc8032 (also including an implementation in Python). However, also these curves are vulnerable to specially crafted input points as showed by Neves and Tibouchi [NT16].

5 SCA Against Post-Quantum Cryptography

NIST is currently standardizing post-quantum cryptography; see overview at csrc.nist.gov/projects/post-quantum-cryptography/selected-algorithms, and has chosen the key-encapsulation mechanisms CRYSTALS Kyber and HQC, and the digital signatures CRYSTALS Dilithium, Falcon, and SPHINCS+. While theoretically secure, there has been an effort to attack and protect these schemes against side-channel attacks; see, e.g., [MGTF19] for an analysis of rejection sampling, number-theoretic transforms, and polynomial multiplications in Dilithium.

NIST recently announced a new call for additional signature schemes, see csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures, and received 40 candidates where 14 of them advanced to the second round. No one has yet analyzed these schemes for "side-channel friendliness".

6 More Advanced SCA with ChipWhisperer

The most common way to protect an implementation against side-channel attacks is through masking; however, this does not protect against glitching [MPO05], and many works have studied how to prove schemes secure against such attacks in the so-called probing model [MMSS19].

Conduct a similar experiment as the "Voltage (VCC) Glitching Raspberry Pi 3 Model B+ with ChipWhisperer-Lite" attack as shown by Colin O'Flynn at youtu.be/dVkCNiM0PL8. We will provide you with a Raspberry Pi 4. Then, use this knowledge to glitch a password checker, an RSA implementation, and/or some other cryptographic scheme.

References

[ABM+02] Adrian Antipa, Daniel Brown, Alfred Menezes, René Struik, and Scott Vanstone. Validation of elliptic curve public keys. In Yvo G. Desmedt, editor, *Public Key Cryptography — PKC 2003*, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[AS20] Jean-Philippe Aumasson and Omer Shlomovits. Attacking threshold wallets. Cryptology ePrint Archive, Paper 2020/1052, 2020. https://eprint.iacr.org/2020/1052.

[BDL+11] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[EPG+20] Andres Erbsen, Jade Philipoom, Jason Gross, Robert Sloan, and Adam Chlipala. Simple high-level code for cryptographic arithmetic: With proofs, without compromises. *SIGOPS Oper. Syst. Rev.*, 54(1), aug 2020.

[Lin21] Yehuda Lindell. Fast secure two-party ecdsa signing. *J. Cryptol.*, 34(4), oct 2021.

[LJL+22] Wenqing Li, Shijie Jia, Limin Liu, Fangyu Zheng, Yuan Ma, and Jingqiang Lin. Cryptogo: Automatic detection of go cryptographic api misuses. In *Proceedings of the 38th Annual Computer Security Applications Conference*, ACSAC '22, New York, NY, USA, 2022. Association for Computing Machinery.

[MGTF19] Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking dilithium. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security*, Cham, 2019. Springer International Publishing.

[MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited: or why proofs in the robust probing model are needed. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(2), Feb. 2019.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked aes hardware implementations. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hard-*

ware and Embedded Systems – CHES 2005, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[MYG23] Nikolaos Makriyannis, Oren Yomtov, and Arik Galansky. Practical key-extraction attacks in leading MPC wallets. Cryptology ePrint Archive, Paper 2023/1234, 2023. https://eprint.iacr.org/2023/1234.

[NT16] Samuel Neves and Mehdi Tibouchi. Degenerate curve attacks. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016*, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[Som16] Juraj Somorovsky. Systematic fuzzing and testing of tls libraries. In *CCS '16*, New York, NY, USA, 2016. Association for Computing Machinery.

[ZBPB17] Jean-Karim Zinzindohoué, Karthikeyan Bhargavan, Jonathan Protzenko, and Benjamin Beurdouche. Hacl*: A verified modern cryptographic library. In *CCS '17*, New York, NY, USA, 2017. Association for Computing Machinery.