NTNU | Norwegian University of Science and Technology

COURSE SUMMARY

TTM4205 – Lecture 18

Tjerand Silde

07.11.2025

Contents

Contents

Presentations

I have sent you an email with the overview of who is presenting when on November 10, 14, and 17. Each of you have 10-15 minutes, and I encourage everyone to ask questions to the other groups after each presentation.

We do not expect you to be done with any of the technical essays, but that you present what the topic you will investigate is about, why you find this interesting, why this is relevant, what you plan to do/write about, and so forth.

Any questions about the presentations?

The Aim of the Course

My goal was to show you a variety of different attacks and mitigations for cryptography systems that we use today. I wanted you to learn how to think as an attacker, so that you better can protect your own schemes going forward.

We went through a lot of material. You are not supposed to remember everything. But you are expected to know what to look for, how to find resources to learn more, have a basic understanding that you can apply to similar issues, and have ideas for how to protect against these attacks.

Contents

Main Takeaways

Main Takeaways

- Security is never better than your entropy source

Main Takeaways

- Security is never better than your entropy source
- Security is based on the best attack against a scheme

Main Takeaways

- Security is never better than your entropy source
- Security is based on the best attack against a scheme
- Today we require 128 bits of security in cryptography

Main Takeaways

- Security is never better than your entropy source
- Security is based on the best attack against a scheme
- Today we require 128 bits of security in cryptography
- We need to ensure access to high entropy randomness

Main Takeaways

- Security is never better than your entropy source
- Security is based on the best attack against a scheme
- Today we require 128 bits of security in cryptography
- We need to ensure access to high entropy randomness
- Pseudorandom Number Generators (PRNGs) expand true randomness into pseudorandom bit streams

Main Takeaways

Main Takeaways

- Most built-in PRNGs are not cryptographically secure

Main Takeaways

- Most built-in PRNGs are not cryptographically secure
- We broke schemes using low-entropy randomness

Main Takeaways

- Most built-in PRNGs are not cryptographically secure
- We broke schemes using low-entropy randomness
- Monte Carlo algorithms are the most efficient way to check if a number is prime, to compute a discrete logarithm or factor a large bi-prime

Main Takeaways

- Most built-in PRNGs are not cryptographically secure
- We broke schemes using low-entropy randomness
- Monte Carlo algorithms are the most efficient way to check if a number is prime, to compute a discrete logarithm or factor a large bi-prime
- We can fool prime-checking if it is not properly randomized

Main Takeaways

- Most built-in PRNGs are not cryptographically secure
- We broke schemes using low-entropy randomness
- Monte Carlo algorithms are the most efficient way to check if a number is prime, to compute a discrete logarithm or factor a large bi-prime
- We can fool prime-checking if it is not properly randomized
- Faulty parameters easily breaks a cryptographic scheme

Contents

Main Takeaways

Main Takeaways

- There is an ongoing debate on regulating cryptography and its effect on privacy, security and safety

Main Takeaways

- There is an ongoing debate on regulating cryptography and its effect on privacy, security and safety
- Many old and weak ciphers are still used today

Main Takeaways

- There is an ongoing debate on regulating cryptography and its effect on privacy, security and safety
- Many old and weak ciphers are still used today
- E.g. MD5, SHA-1, RC4, 3DES are not fully revoked yet

Main Takeaways

- There is an ongoing debate on regulating cryptography and its effect on privacy, security and safety
- Many old and weak ciphers are still used today
- E.g. MD5, SHA-1, RC4, 3DES are not fully revoked yet
- Export ciphers leading to weak parameters for DH and RSA

Main Takeaways

- There is an ongoing debate on regulating cryptography and its effect on privacy, security and safety
- Many old and weak ciphers are still used today
- E.g. MD5, SHA-1, RC4, 3DES are not fully revoked yet
- Export ciphers leading to weak parameters for DH and RSA
- Use ephemeral elliptic curve DH for key exchange

Main Takeaways

- There is an ongoing debate on regulating cryptography and its effect on privacy, security and safety
- Many old and weak ciphers are still used today
- E.g. MD5, SHA-1, RC4, 3DES are not fully revoked yet
- Export ciphers leading to weak parameters for DH and RSA
- Use ephemeral elliptic curve DH for key exchange
- DualEC and standardized schemes with backdoors

Contents

Main Takeaways

Main Takeaways

- Error messages can leak important information

Main Takeaways

- Error messages can leak important information
- Padding checks can leak important information

Main Takeaways

- Error messages can leak important information
- Padding checks can leak important information
- Adaptive decryption queries can exploit this

Main Takeaways

- Error messages can leak important information
- Padding checks can leak important information
- Adaptive decryption queries can exploit this
- AES-CBC is only IND-CPA secure, not IND-CCA

Main Takeaways

- Error messages can leak important information
- Padding checks can leak important information
- Adaptive decryption queries can exploit this
- AES-CBC is only IND-CPA secure, not IND-CCA
- AES-CBC is removed in TLS 1.3 to avoid attacks

Main Takeaways

- Error messages can leak important information
- Padding checks can leak important information
- Adaptive decryption queries can exploit this
- AES-CBC is only IND-CPA secure, not IND-CCA
- AES-CBC is removed in TLS 1.3 to avoid attacks
- AES-CBS and RSA-PKCS#1v1.5 are vulnerable

Main Takeaways

- Error messages can leak important information
- Padding checks can leak important information
- Adaptive decryption queries can exploit this
- AES-CBC is only IND-CPA secure, not IND-CCA
- AES-CBC is removed in TLS 1.3 to avoid attacks
- AES-CBS and RSA-PKCS#1v1.5 are vulnerable
- Efficiency depends on how strict checks are

Main Takeaways

Main Takeaways

- Use authenticated mode of AES (AEAD)

Main Takeaways

- Use authenticated mode of AES (AEAD)
- Be vary of length extension attacks against SHA-2

Main Takeaways

- Use authenticated mode of AES (AEAD)
- Be vary of length extension attacks against SHA-2
- Do not use RSA encryption unless you really have to

Main Takeaways

- Use authenticated mode of AES (AEAD)
- Be vary of length extension attacks against SHA-2
- Do not use RSA encryption unless you really have to
- If you have to, then use RSA-OAEP padding

Main Takeaways

- Use authenticated mode of AES (AEAD)
- Be vary of length extension attacks against SHA-2
- Do not use RSA encryption unless you really have to
- If you have to, then use RSA-OAEP padding
- We studied the Bleichenbacher attack

Main Takeaways

- Use authenticated mode of AES (AEAD)
- Be vary of length extension attacks against SHA-2
- Do not use RSA encryption unless you really have to
- If you have to, then use RSA-OAEP padding
- We studied the Bleichenbacher attack
- Use encrypt-then-authenticate if possible

Contents

Main Takeaways

Main Takeaways

- Quantum computers can break schemes based on factoring and DL

Main Takeaways

- Quantum computers can break schemes based on factoring and DL
- We need to design new schemes based on other hardness assumptions

Main Takeaways

- Quantum computers can break schemes based on factoring and DL
- We need to design new schemes based on other hardness assumptions
- We looked at the quantum-safe lattice assumptions LWE and SIS

Main Takeaways

- Quantum computers can break schemes based on factoring and DL
- We need to design new schemes based on other hardness assumptions
- We looked at the quantum-safe lattice assumptions LWE and SIS
- We studied the new quantum-safe standards ML-KEM and ML-DSA

Main Takeaways

- Quantum computers can break schemes based on factoring and DL
- We need to design new schemes based on other hardness assumptions
- We looked at the quantum-safe lattice assumptions LWE and SIS
- We studied the new quantum-safe standards ML-KEM and ML-DSA
- They are in general fast but keys, ciphertexts and signatures are large

Contents

Main Takeaways

Main Takeaways

- Black-box crypto, Kerckhoff's principle, implementation

Main Takeaways

- Black-box crypto, Kerckhoff's principle, implementation
- Leakage such as timings, power consumption, radiation, temperature, memory patterns, sound, ...

Main Takeaways

- Black-box crypto, Kerckhoff's principle, implementation
- Leakage such as timings, power consumption, radiation, temperature, memory patterns, sound, ...
- Examples: credit cards, shared resources, malware,...

Main Takeaways

- Black-box crypto, Kerckhoff's principle, implementation
- Leakage such as timings, power consumption, radiation, temperature, memory patterns, sound, ...
- Examples: credit cards, shared resources, malware,...
- Remote vs physical, and software vs hardware attacks

Main Takeaways

- Black-box crypto, Kerckhoff's principle, implementation
- Leakage such as timings, power consumption, radiation, temperature, memory patterns, sound, ...
- Examples: credit cards, shared resources, malware,...
- Remote vs physical, and software vs hardware attacks
- Passive vs active, and invasive vs non-invasive attacks

Main Takeaways

- Black-box crypto, Kerckhoff's principle, implementation
- Leakage such as timings, power consumption, radiation, temperature, memory patterns, sound, ...
- Examples: credit cards, shared resources, malware,...
- Remote vs physical, and software vs hardware attacks
- Passive vs active, and invasive vs non-invasive attacks
- Constant time code, randomization, fault protection,...

Main Takeaways

Main Takeaways

- Square-and-multiply must be regular and randomized

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder
- Integer arithmetic such as IMUL must be constant time

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder
- Integer arithmetic such as IMUL must be constant time
- Modular addition and reduction must be constant time

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder
- Integer arithmetic such as IMUL must be constant time
- Modular addition and reduction must be constant time
- Modular inversion must also be constant time

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder
- Integer arithmetic such as IMUL must be constant time
- Modular addition and reduction must be constant time
- Modular inversion must also be constant time
- We use universal curve-dependent formulas for ECC

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder
- Integer arithmetic such as IMUL must be constant time
- Modular addition and reduction must be constant time
- Modular inversion must also be constant time
- We use universal curve-dependent formulas for ECC
- We can use bit-slicing and masking to protect AES

Main Takeaways

- Square-and-multiply must be regular and randomized
- We studied how to implement Montgomery Ladder
- Integer arithmetic such as IMUL must be constant time
- Modular addition and reduction must be constant time
- Modular inversion must also be constant time
- We use universal curve-dependent formulas for ECC
- We can use bit-slicing and masking to protect AES
- Similar techniques applies to post-quantum cryptography

Contents

Main Takeaways

Main Takeaways

- Must enforce honest behavior in protocols

Main Takeaways

- Must enforce honest behavior in protocols
- Must verify correctness of parameters and inputs

Main Takeaways

- Must enforce honest behavior in protocols
- Must verify correctness of parameters and inputs
- Must avoid corner case leakage and replay attacks

Main Takeaways

- Must enforce honest behavior in protocols
- Must verify correctness of parameters and inputs
- Must avoid corner case leakage and replay attacks
- Must always verify output values for faults

Contents

Main Takeaways

Main Takeaways

- Commitments: binding and hiding

Main Takeaways

- Commitments: binding and hiding
- ZK Proofs: sound and zero-knowledge

Main Takeaways

- Commitments: binding and hiding
- ZK Proofs: sound and zero-knowledge
- Pedersen and ElGamal commitment backdoors

Main Takeaways

- Commitments: binding and hiding
- ZK Proofs: sound and zero-knowledge
- Pedersen and ElGamal commitment backdoors
- ZKPs can be faked if we do not hash everything

Main Takeaways

- Commitments: binding and hiding
- ZK Proofs: sound and zero-knowledge
- Pedersen and ElGamal commitment backdoors
- ZKPs can be faked if we do not hash everything
- The Schnorr signature is a ZKP of discrete log

Contents

From what I can see, you have learned a lot and performed well this semester. I am certain that the way of thinking, our discussions, and the problems you have solved in this course will be useful for all of you going forward.

I hope that you enjoyed the course, that it was challenging but interesting, and that you see the value of your effort. I hope your assignments goes well!

Questions?