NTNU | Norwegian University of Science and Technology

COURSE SUMMARY

TTM4205 – Lecture 17

Tjerand Silde

08.11.2024

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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.

Course Content

The course covers how to implement, analyse, attack, protect and securely compose cryptographic algorithms in practice. It goes in depth on how to

- implement computer arithmetic
- attack implementations using side-channel attacks and fault injection
- exploit padding oracles and low-entropy randomness
- utilise techniques to defend against these attacks
- securely design misuse-resistant APIs

Learning Outcome

Knowledge

Advanced knowledge about the mathematical building blocks underlying modern cryptography, properties of and applications of cryptographic primitives, challenges and common mistakes when implementing cryptography, side-channel attacks and countermeasures, and high level design principles for secure use of cryptography in practice.

Learning Outcome

Skills

Able to implement the underlying mathematics and high-level protocols used in symmetric key and public key cryptosystems, perform simple side-channel attacks and implement countermeasures, analyse side-channel countermeasures and design misuse resistant APIs for cryptography.

Guest Lectures

We have two upcoming guest lectures in this course:

- Tuesday November 12 at 08:15-10:00: Håkon Jacobsen (Thales Norway) on “FPGAs and Cryptography”
- Friday November 15 at 10:15-12:00: Hans Heum (NTNU) on “Quantum Computers and Cryptography”

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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
- The most efficient algorithms for checking if a number is prime, to compute a discrete logarithm or factor a large bi-prime are all randomized Monte Carlo algorithms

Main Takeaways

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

Main Takeaways

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

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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

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
- 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
- Use ephemeral elliptic curve DH for key exchange
- DualEC and standardized schemes with backdoors

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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 CPA secure, not 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 CPA secure, not 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 CPA secure, not 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 CPA secure, not 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

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 must

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 must
- If you must, 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 must
- If you must, 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 must
- If you must, then use RSA-OAEP padding
- We studied the Bleichenbacher attack
- Use encrypt-then-authenticate if possible

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

Main Takeaways

Main Takeaways

- Consistency of keygen and parameters matters

Main Takeaways

- Consistency of keygen and parameters matters
- How schemes (AES+RSA) are composed matters

Main Takeaways

- Consistency of keygen and parameters matters
- How schemes (AES+RSA) are composed matters
- We need very concise protocol descriptions

Main Takeaways

- Consistency of keygen and parameters matters
- How schemes (AES+RSA) are composed matters
- We need very concise protocol descriptions
- Always (try to) prove security of a protocol

Main Takeaways

- Consistency of keygen and parameters matters
- How schemes (AES+RSA) are composed matters
- We need very concise protocol descriptions
- Always (try to) prove security of a protocol
- Make code open source and pay for audits

Main Takeaways

Main Takeaways

- Use domain separation for similar functions

Main Takeaways

- Use domain separation for similar functions
- Have integrity checks for all messages

Main Takeaways

- Use domain separation for similar functions
- Have integrity checks for all messages
- Do not re-use keys across applications

Main Takeaways

- Use domain separation for similar functions
- Have integrity checks for all messages
- Do not re-use keys across applications
- Do not design your own schemes / protocols

Main Takeaways

- Use domain separation for similar functions
- Have integrity checks for all messages
- Do not re-use keys across applications
- Do not design your own schemes / protocols
- Use up-to-date modern primitives and libraries

Contents

General Information

Randomness

Legacy Crypto

Side-Channel Attacks

Padding Oracles

Protocols APIs

Commitments and Zero-Knowledge

Protocol Composition

Final Thoughts

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.

Questions?