MATH-489

Number theory in cryptography

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cybersecurity	MA1	Opt.
Ing.-math	MA1, MA3	Opt.
Mathematics for teaching	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Written
Workload	150h
Weeks	14
Hours	**4 weekly**
Courses	2 weekly
Exercises	2 weekly
Number of positions	

Remark

Cours donné en alternance tous les 2 ans (pas donné en 2018-19)

Summary

The goal of the course is to introduce basic notions from public key cryptography (PKC) as well as basic number-theoretic methods and algorithms for cryptanalysis of protocols and schemes based on PKC.

Content

Basic notions and algorithms from public key cryptography such as RSA, ElGamal, key exchange protocols, zero knowledge proofs. Main topics may include, but are not limited to

- modular and finite field arithmetic
- primality testing
- polynomial and integer factorization algorithms
- index calculus and discrete logarithm-based schemes
- elliptic curve cryptography
- basic notions from lattice-based cryptography

Keywords

public key cryptography, key exchange, digital signatures, zero knowledge proofs, RSA, ElGamal, integer factorization, index calculus, elliptic curve cryptography

Teaching methods

lecture notes, additional references

Assessment methods

Theoretical assignments: Weekly problem sets focusing on number-theoretic and complexity-theoretic aspects. Theoretical assignments will count for 30% of the final grade.

Programming assignments: All of the programming exercises will be in SAGE which is a Python-based computer algebra system. No prior experience with SAGE or Python is required. Programming assignments will count for 30% of the final grade.
One mid-term exam (15% of the final grade) and **one final exam** (25% of the final grade). Both exams will test theoretical understanding as well as understanding of the algorithms and protocols. The exams will include no SAGE programming exercises. If needed, algorithms could be presented with pseudo-code.