Advanced Cryptography
MFOCS, Oxford

Christophe Petit

January 16, 2018

Contents

1 Introduction

1.1 Course Content and Prerequisites

Cryptography is the science and art of ensuring private and authenticated communications. How does modern cryptography proceed to achieve that?

- We provide rigorous definitions of what security means in a given context, for example IND-CCA security or existential unforgeability definitions.
- We make some hardness assumption on some computational problem, for example the discrete logarithm problem or the integer factorization problem.
- We build some protocol so that we can prove that breaking the protocol (in the sense of our security definition) would imply solving the hard computational problem. For example, ElGamal encryption is IND-CPA secure if the decisional Diffie-Hellman problem is hard.

In this course I will assume you have the basic knowledge on cryptography that is normally provided in an introduction course, and certainly the one offered in MFOCS. I also assume that finite fields have no secret for you, as well as basic algebra concepts such as rings and groups. In addition to this, I assume that you have some experience with Sage and know some basic computer algebra algorithms.

Of course, I assume that you are motivated, eager to learn and hard-working. I hope you will feel comfortable to ask questions when we meet or by email anytime.

Compared to the introduction course offered in MFOCS, this one is likely to focus more on the mathematical constructs and less on the security definitions. This does not mean that I do not consider those as important. I assume you master the ones covered in the introduction course, and we will also study some new definitions.

Based on the preferences you have expressed, we will cover the following content this year:

1. Elliptic curve cryptography, including isogeny-based cryptography.
2. Zero-knowledge protocols, including their application to voting protocols.

I can provide references on other topics of your interest upon request, but these won't be considered part of the assessed material. I am also open to discuss potential dissertations topics.

1.2 Organisation

This course will be organized as a reading course.

You are expected to learn by yourselves or in groups from the reference material. The questions listed in Section 2 should help you identify the basic knowledge to acquire on each of the topics listed above.

The questions listed in Section 3 cover advanced material or they aim at developing your critical understanding. You are required to prepare a 60 min group presentation answering them. The presentation will be interrupted by discussions to check your understanding of the material covered, or to provide additional information. You are welcome to split the work between you, but everybody must have a good understanding of the whole content. Your understanding of this material may be assessed in the mini-project.

Here is a tentative schedule **(to be agreed)**, per week

1. Tuesday 16/01, 12pm-13pm: introduction
2. Tuesday 23/01, 11am-12pm: discussion on reading material (if requested)
3. Tuesday 30/01, 9am-10am and 11am-13pm: 2h for presentations + 1h for questions/discussions on reading material
4. Tuesday 06/02, 11am-12pm: discussion on reading material (if requested)
5. Tuesday 13/02, 9am-10am and 11am-13pm: 2h for presentations + 1h for questions/discussions on reading material
6. Tuesday 20/02, 11am-12pm: discussion on reading material (if requested)
7. **Wednesday** 28/02, **to be agreed**: 2h for presentations + 1h for questions/discussions on reading material
8. Tuesday 06/03, 11am-12pm: discussion on reading material (if requested)

By "requested" I mean that at least one of you has explicitly asked me to meet. All meetings will be open to everyone. We may occasionally have a meeting over Skype (not when you will be presenting).

1.3 Feedback welcome

I have been teaching Advanced Cryptography for MFOCS since 2015 as a regular course, but this year I will teach it as a reading course for the first time. I am very keen to improve it, and your constructive feedback can help me with that. Feel free to provide feedback any time and be ensured it will be very welcome!

2 Guided Reading

2.1 Elliptic Curve Cryptography

2.1.1 Elliptic Curves (Week 1)

The main references for this part are my lecture slides from last year [31] and Silverman's book [35], Chapter III.

- What are the advantages of elliptic curve cryptography compared to cryptography based on the discrete logarithm over finite fields? Look at `www.keylength.com`
- Define an elliptic curve.
- Define Weierstrass equations. Can any elliptic curve be represented by a Weierstrass equation?
- Define reduced Weierstrass equations. Why are the characteristic 2 and 3 cases treated separately?
- Define the discriminant of an elliptic curve. How is this related to the smoothness condition?
- Define the j-invariant of an elliptic curve. In what sense is this an invariant?
- Suppose $\varphi : E_1 \to E_2$ is an isomorphism mapping one curve in Weierstrass coordinates to another curve in Weierstrass coordinates. What is the most general form of φ?
- Give the equation of one curve with j-invariant j.
- Define a group law on the points of an elliptic curve. Assuming the curve is in (reduced) Weierstrass coordinates, how do you add two points together? What is the neutral element? How do you define the inverse of a point? Understand the group operation both from geometric and algebraic points of view.
- Prove that this operation indeed defines a group. The only difficult property to prove is associativity. I personally like Sutherland's proof in his MIT lecture notes (`http://math.mit.edu/classes/18.783/2017/LectureNotes2.pdf`)
- Define scalar multiplication. Express scalar multiplication as a rational map.
- Define torsion points and division polynomials. How are these related?
- What is the group structure of elliptic curves over finite fields? How does it look like over their algebraic structure?
- Understand Hasse's theorem. For any finite field K, why do we expect to have roughly K points on any curve defined over K? How accurate is this prediction in general?
- Understand Weil-Deligne's theorem. Assume you know the number of points of a curve over some finite field. How can you use this theorem to deduce the number of points over any extension field?

- Define an isogeny. Show that its kernel is a subgroup of the curve. Define the dual isogeny.
- Define an endomorphism. Define the Frobenius endomorphism. What is its characteristic equation?
- Describe the structure of endomorphisms.
- Use Sage: define a finite field, an elliptic curve, add points on the curve, perform scalar multiplications, compute the number of rational points.

2.1.2 Elliptic Curve Discrete Logarithm Problem (Weeks 1-2)

The main references for this part are my lecture slides from last year [31], and Blake-Serousi-Smart [6].

- Define the elliptic curve discrete logarithm problem (ECDLP)
- Define Computational/ decisional elliptic curve Diffie-Hellman problems
- Define the EC Diffie-Hellman protocol. What security guarantees does this protocol offer?
- Define EC ElGamal. Show that ElGamal is IND-CPA secure if ECDDH is hard. Show that ElGamal is not IND-CCA secure.
- Define ECDSA. What security guarantees does this protocol offer?
- Discuss the importance of using good randomness in ECDSA (attack on Sony's signatures, bitcoin theft due to Android's RNG weaknesses)
- Compare existing attack on DLP and ECDLP.
- What are NIST curves? Are these curves perfectly safe to use?
- Check othe popular curves on `ttp://safecurves.cr.yp.to/`
- There are two main methods to generate suitable curves: the first one is to generate random coefficients and use point counting algorithms; the second one is the complex multiplication method. Discuss the advantages of each approach.
- Describe Schoof's point counting algorithm, and discuss its complexity.
- What is the quadratic twist of a curve? How are the number of points on one curve and its quadratic twist related?
- Use Sage to implement EC ElGamal encryption algorithm and Pollard's rho algorithm on ECDLP.

2.1.3 Algorithmic and implementation aspects (Week 2)

The main references for this part are my lecture slides from last year [31], and Blake-Serousi-Smart [6].

- Describe the basic double-and-add algorithm. What is its complexity?
- Study various methods to accelerate scalar multiplication [6, Chapter 4].
- Define Edwards curves. Are they as generic as Weierstrass curves? What are the advantages of using Edwards curves formulae over Weirstrass curves formulae?
- Use Sage to implement a scalar multiplication method of your choice for Edwards curves.

2.1.4 Pairing-based cryptography (Week 3)

The main references for this part are my lecture slides from last year [31], the survey [17], and references therein.

- Define pairings. What are their main properties? What are Type I, Type II, Type III pairings?
- How can pairings help to build a 3-partite Diffie-Hellman protocol? Can this protocol be built with any type of pairing?
- What computational assumptions are required for the security of the 3-partite Diffie-Hellman protocol? How can these assumptions be related to other computational assumptions?
- Can we extend this protocol to a 4-partite Diffie-Hellman protocol?
- What is the main idea behind identity-based cryptography? What problem does it aim to solve and what new problems does it create?
- Study Boneh-Franklin ID-based encryption protocol. Check that decryption of a valid ciphertext gives back the corresponding plaintext.
- What are the security requirements for ID-based encryption? Do the requirements posed by Boneh-Franklin look sufficient to you?
- What are the security requirements for the hash function in Boneh-Franklin protocol?
- Study Boneh-Lynn-Sacham signatures. What security arguments/proofs can be provided for this protocol?
- Use Sage to implement one pairing-based protocol of your choice.

2.1.5 Elliptic curve pairings (Week 3)

The main references for this part are my lecture slides from last year [31], and Galbraith's survey in Blake-Serousi-Smart [6][Vol 2, Chapter IX].

- Define a divisor, its degree, its support.
- Define the divisor of a function, a principal divisor
- Show that the map sending an elliptic curve point P to the divisor $(P)-(0)$ is a group homomorphism up to principal divisors
- Let $D = \sum_P n_P(P)$ be a degree 0 divisor on E. Then $D \sim 0$ if and only if $\sum_P [n_P]P = O$.
- State the Weil reciprocity.
- Define the Tate pairing. State and prove its main properties.
- Define the reduced Tate pairing. Why is this definition useful?
- Define the embedding degree. What is its importance for cryptographic applications? What is expected about the embedding degree of random curves?
- Define the Weil pairing. State and prove its main properties.
- Show how the Tate and Weil pairings reduce ECDLP to DLP over a finite field. Does this reduction necessarily give a subexponential algorithm for ECDLP?
- Can we use the Tate and Weil pairings for protocols that require a symmetric pairing?
- What is a distorsion map and how can it help buidling a symmetric pairing?
- Study how the Tate and Weil pairing can be efficiently computed using Miller's algorithm
- What are the requirements for elliptic curves to have suitable cryptographic pairings? Give some examples of suitable families.

2.1.6 Isogeny-based cryptography (Week 4)

The main references for this part are my lecture slides from last year [31, **?**] and the recent survey [18].

- Define an isogeny; give a standard representation as a rational map. Define the kernel, the degree. Define the dual isogeny.

- What are Vélu's formulae? In general, what is the computational cost of these formulae?
- Define the endomorphism ring computation problem. How should the answer to this problem be returned? Is this representation efficient in general?
- Define isogeny graphs and state their main properties.
- Sketch Kohel's algorithm to compute endomorphisms of supersingular elliptic curves. What is its cost?
- What is the isogeny computation problem?
- Why are isogeny problems appealing for cryptography?
- Describe Charles-Goren-Lauter hash function based on isogenies. What are the security arguments for this function? What is known about its security?
- Describe the supersingular isogeny key exchange protocol (SIDH). Why are Alice and Bob's secret kernels of order coprime to each other? Could we complete the protocol without exchanging extra points (i.e. exchanging only j-invariants)? What are the security guarantees provided by the protocol?
- Show how to derive an encryption scheme from this key exchange protocol.
- Implement CGL hash function in Sage.

2.2 Zero-knowledge protocols

As a warm-up, read the fabulous story of crypto researcher Mick Ali, a descendant of Ali Baba who wanted to prove his knowledge of the magic word without revealing it [33].

Zero-knowledge protocols are subtle cryptographic primitives that are often used in bigger protocols, and there is unfortunately no definitive reference for their study. There is some consensus on what the security definitions of these protocols should convey, but there are also many subtle variations on the exact definitions in the literature. These variations are due to the necessity to adjust the definitions to either what can be proven from the particular construction or to what should be proven for the bigger protocol, and ideally both simultaneously.

Besides the early fundamental works including [21, 20, 8], good introductions can be found in Chapter 8 of Katz [25] and the lecture notes of Damgård [13] and Venturi [37].

2.2.1 Basic definitions (Week 5)

The main references here are the lecture notes of Damgård [13] and Venturi [37].

- Define language, NP language, relation, witness.
- How do interactive proofs differ from classical proofs in Maths textbooks?
- Define sigma protocols, prover, verifier, commitment, challenge, response. Informally define the correctness, soundness and zero-knowledge properties of sigma protocols.
- Define k-special soundness. What is the intuition behind this security definition?
- Define honest verifier zero-knowledge. What is the intuition behind this security definition?
- Explain the difference between arguments and proofs.
- Explain the difference in the notions of perfect/statistical/computational soundness and zero-knowledge.
- What are the properties satisfied by Mick Ali's protocol? [33]

2.2.2 Some classical proofs (Week 5)

The main references here is the paper [20].

- Study the zero-knowledge proof for graph isomorphism in [20, Section 2]. What are the properties satisfied by this protocol?
- Study the zero-knowledge proof for 3-coloring in [20, Section 3]. What are the properties satisfied by this protocol?
- Explain how this leads to a zero-knowledge proof for any language in NP. What are the properties satisfied by this protocol?
- Study how the "encryption scheme" is used in these protocols. Can you replace this encryption scheme by a one-way function?
- Study how to perform "OR proofs" in zero-knowledge [37, 12].

2.2.3 The Fiat-Shamir transform (Week 6)

- Define identification protocol. What are the security requirements of such protocols?
- Study the so-called "Fiat-Shamir transform" [15, 32]. What is its purpose? What is the underlying rationale?

- What are the typical requirements on the hash function in the Fiat-Shamir transform? What is the purpose of the so-called "forking lemma" [32]? Understand this lemma and its use.
- Study Schnorr identification scheme. Check that it satisfies all the above security requirements. How are Schnorr signatures constructed from the identification scheme?
- Can the above requirement on the hash function be relaxed? Study the arguments given in [30].
- Suppose an identification scheme is such that a malicious prover can cheat with a probability 1/2. What can be done to reduce this probability?
- Define an elliptic curve version of Schnorr signatures, and implement it using Sage.

2.2.4 E-voting (Weeks 7-8)

The main reference for this part is Benaloh's and Adida's papers [5, 2].

- Reflect on voting and electronic voting. What are the security requirements that you can identify? Which ones were/not satisfied by the voting processes you have used in the past? Which security requirements do you consider as the most important ones?
- Compare your list of security requirements with the ones identified in [1].
- Recall ElGamal encryption protocol. Show how ElGamal ciphertext can be re-randomized without the secret key.
- What is a treshold encryption protocol?
- Explain how to turn ElGamal encryption into a treshold encryption protocol.
- Describe Chaum-Pedersen's zero-knowledge protocol to prove discrete logarithm equalities, and explain how to use this protocol to prove correct decryption of ElGamal [2, 9]. Verify that this protocol satisfies the standard security requirements.
- Define mix-net and shuffle.
- Describe the Sako-Kilian shuffle proof [34, 2].
- What is the purpose of separating ballot creation from ballot casting?
- What is the purpose of auditing in both Benaloh and Adida's systems? Who is supposed to perform this audit? What are the technical skills required from auditors? How realistic is it that auditing will be properly performed?

- What are the main lessons taken from the electronic voting experiment described in [3]? What were the main technical and non technical challenges solved for this election?
- Briefly describe and compare the the voting protocols in [5, 2, 3]. Comment on the type of elections for which those protocols could be suitable.
- Can electronic voting provide additional security properties compared to traditional voting techniques? Can traditional voting techniques provide additional security properties compared to electronic voting?
- Describe the shuffle argument in [4] and implement it in Sage.

2.2.5 Further reading: ZK proofs based on other assumptions

For those interested, I include in this section references to zero-knowledge proofs based on a number of different assumptions.

- Code-based cryptography: the classic reference is Stern [36].
- Lattice-based cryptography: see [26] for a Stern-like proof, or Lyubashevsky [29].
- Isogeny-based cryptography: see [14] or my own paper [18].

3 Group Presentations

3.1 Elliptic Curve Cryptography (I) (Week 3)

1. Read the survey on recent elliptic curve discrete logarithm results [19]. Summarize the current state of the art and main open problems.
2. Describe the Elliptic Curve Factorization Method. What is its complexity? Is it used today to factor big numbers? Implement the method using Sage. Compare the efficiency of your implementation with the internal routine.
3. Study Golwasser-Killian primality proofs. Implement the method in Sage. Study the efficiency of your implementation.
4. Describe Ciet-Joye fault's attack [10]. What is the attack model? What can be achieved with this attack?

Prepare a presentation to show your results at the beginning of Week 3.

3.2 Elliptic Curve Cryptography (II) (Week 5)

1. Pairings have resulted in a tremendous amount of cryptographic schemes. Browse the web to find some applications and list them into categories. Choose one application that has not been covered in the lecture slides and describe it.

2. Read Koblitz-Menezes' opinion on the proliferation of pairing assumptions [28] and summarize their main arguments.

3. Based on the papers [16, 17], provide constructions of elliptic curve pairings that are suitable for the protocol chosen in Question 1. Explain the rationales behind your choices, and discuss the resulting security and efficiency both asymptotically and at the 128-bit security level. In your parameter estimation, be sure to take into account recent progress in discrete logarithm computation [27]. How is the discussion in Q2 relevant here?

4. Implement Miller's algorithm in Sage. Compare the cost of a pairing and a scalar multiplication for your implementation. Does this fit with the a priori theoretical estimations?

Prepare a presentation to show your results at the beginning of Week 5.

3.3 Zero-knowledge protocols (Week 7)

1. While the early works have shown the existence of zero-knowledge proofs for all languages in NP, subsequent work has dramatically improved the efficiency. Sketch the proofs for arithmetic circuits proposed in [11, 23, 22]. What is the motivation for working with arithmetic circuits instead of 3-coloring or SAT problems? Are efficiency improvements only due to new clever ideas or do they come at the cost of relaxed security definitions, stronger assumptions? How is [7] improving on [22]?

2. For specific classes of languages one can improve efficiency even further. Describe Groth-Sahai proofs [24]. What sort of statements can they prove? What security assumptions are required for that?

3. Choose one of the protocols above and prove that it satisfies its main security requirements.

4. Implement one of the above proofs of your choice using Sage. Propose suitable parameters to use in your implementation.

Prepare a presentation to show your results at the beginning of Week 7.

References

[1] Ben Adida. Lecture notes: Special topics in cryptography, by ran canetti and ron rivest. `http://courses.csail.mit.edu/6.897/spring04/materials.html`, 2004.

[2] Ben Adida. Helios: Web-based open-audit voting. In *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*, pages 335–348, 2008.

[3] Ben Adida, Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. In T. Moran D. Jefferson, J.L. Hall, editor, *Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. Usenix, 8 2009.

[4] Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 263–280. Springer, 2012.

[5] Josh Benaloh. Simple verifiable elections. In *2006 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'06, Vancouver, BC, Canada, August 1, 2006*, 2006.

[6] Ian F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography*. Cambridge University Press, New York, NY, USA, 1999.

[7] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2016.

[8] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[9] David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1992.

[10] Mathieu Ciet and Marc Joye. Elliptic curve cryptosystems in the presence of permanent and transient faults. *IACR Cryptology ePrint Archive*, 2003:28, 2003.

[11] Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 424–441, 1998.

[12] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, volume 893 of *Lecture Notes in Computer Science*, pages 174–187, 1994.

[13] Ivan Damgård. On σ-protocols. Lecture Notes, University of Aarhus, Department for Computer Science, 2010.

[14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Mathematical Cryptology*, 8(3):209–247, 2014.

[15] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[16] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves, 2006.

[17] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[18] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, pages 3–33, 2017.

[19] Pierrick Gaudry and Steven D. Galbraith. Recent progress on the elliptic curve discrete logarithm problem. To appear in Design, Codes and Cryptography, 2015.

[20] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, 1991.

[21] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[22] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 192–208. Springer, 2009.

[23] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358. Springer, 2006.

[24] Jens Groth and Amit Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.

[25] Jonathan Katz. *Digital Signatures*. Springer, 2010.

[26] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.

[27] Taechan Kim and Razvan Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 543–571, 2016.

[28] Neal Koblitz and Alfred J. Menezes. The brave new world of bodacious assumptions in cryptography. *Notices of the American Mathematical Society*, 57:357–365, 2010.

[29] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*, pages 162–179, 2008.

[30] Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. Hash function requirements for schnorr signatures. *J. Mathematical Cryptology*, 3(1):69–87, 2009.

[31] Christophe Petit. Elliptic curve cryptography. Lecture slides for MFOCS, oxford, 2016.

[32] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[33] Jean-Jacques Quisquater, Louis Guillou, Marie Annick, and Tom Berson. How to explain zero-knowledge protocols to your children. In *Proceedings on Advances in Cryptology*, CRYPTO '89, pages 628–631, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[34] Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - A practical solution to the implementation of a voting booth. In *Advances in Cryptology - EUROCRYPT '95, International Conference on the Theory*

and Application of Cryptographic Techniques, Saint-Malo, France, May 21-25, 1995, Proceeding, pages 393–403, 1995.

[35] Joseph Silverman. *The Arithmetic of Elliptic Curves*. Springer Verlag, 1986.

[36] Jacques Stern. A new paradigm for public key identification. *IEEE Trans. Information Theory*, 42(6):1757–1768, 1996.

[37] Daniele Venturi. Zero-knowledge proofs and applications. Lecture Notes, University of Rome, 2015.