

PETs H25

Exercise Session 3

David Pulido

PETs Student TA

07. October 2025, Zurich



Outline

1. Further Motivation (3 min)
2. Reminder: Extra Resources (2 min)
3. Refresher on ZKP's (Simulators and Extractors, 10 min)
4. Exercises (30 min)
5. Questions & Feedback

Further Motivation

- Building Block for Post-Quantum Crypto
- E-Cash
- Machine Learning
- Differential Privacy Auditing (see second half of course)

Resources (Unfinished)

A Graduate Course in Applied Cryptography

Dan Boneh and Victor Shoup

Version 0.6, Jan. 2023

A Basic number theory	1096
A.1 Cyclic groups	1096
A.2 Arithmetic modulo primes	1096
A.2.1 Basic concepts	1096
A.2.2 Structure of \mathbb{Z}_p^*	1097
A.2.3 Quadratic residues	1097
A.2.4 Computing in \mathbb{Z}_p	1098
A.2.5 Summary: arithmetic modulo primes	1099
A.3 Arithmetic modulo composites	1099
B Basic probability theory	1101
B.1 The birthday Paradox	1101
B.1.1 More collision bounds	1103
B.1.2 A simple distinguisher	1103
C Basic complexity theory	1105
D Probabilistic algorithms	1106

20 Proving properties in zero-knowledge	823
20.1 Languages and soundness	823
20.2 Proving properties on encrypted data	824
20.2.1 A generic protocol for non-linear relations	829
20.3 Non-interactive proof systems	831
20.3.1 Example: a voting protocol	831
20.3.2 Non-interactive proofs: basic syntax	833
20.3.3 The Fiat-Shamir transform	833
20.3.4 Non-interactive soundness	834
20.3.5 Non-interactive zero knowledge	834
20.3.6 An example: applying the Fiat-Shamir transform to the Chaum-Pedersen protocol	837
20.4 Computational zero-knowledge and applications	838
20.4.1 Example: range proofs	839
20.4.2 Special computational HVZK	840
20.4.3 An unconstrained generic protocol for non-linear relations	841
20.5 Bulletproofs: compressed Sigma protocols	842
20.6 Succinct non-interactive zero-knowledge proofs (SNARKs)	842
20.7 A fun application: everything that can be proved, can be proved in zero knowledge	842
20.8 Notes	842
20.9 Exercises	843

Resources

INTRODUCTION TO MODERN CRYPTOGRAPHY

Second Edition

Jonathan Katz

Appendix A Mathematical Background	537
A.1 Identities and Inequalities	537
A.2 Asymptotic Notation	537
A.3 Basic Probability	538
A.4 The “Birthday” Problem	542
A.5 *Finite Fields	544
 Appendix B Basic Algorithmic Number Theory	 547
B.1 Integer Arithmetic	549
B.1.1 Basic Operations	549
B.1.2 The Euclidean and Extended Euclidean Algorithms	550
B.2 Modular Arithmetic	552
B.2.1 Basic Operations	552
B.2.2 Computing Modular Inverses	552
B.2.3 Modular Exponentiation	553
B.2.4 *Montgomery Multiplication	556
B.2.5 Choosing a Uniform Group Element	557
B.3 *Finding a Generator of a Cyclic Group	559
B.3.1 Group-Theoretic Background	559
B.3.2 Efficient Algorithms	561
References and Additional Reading	562
Exercises	562

ZKP – Simulator

- Computational knowledge: ability to compute something efficiently (e.g. knowing the answers of PETs homework let's you solve the problem sheet quickly)
- If the verifier could have faked the same conversation alone, they learned nothing from the real one → no new ability for efficient computation was gained

ZKP – Simulator

Sim(G)

Guess $\hat{c} \leftarrow \{0, 1\}$

If $\hat{c} = 0$, commit to a random permutation Π of G

If $\hat{c} = 1$, commit to a complete graph

Send the commitments to the verifier who returns c

If $\hat{c} \neq c$, abort and restart

Else, open the commitments as requested by the verifier

Output the view $(G, \text{Commit}, c, \text{Open})$

$P(G, \text{Ham-Cycle})$

$V(G)$

pick a random permutation Π
of the n vertices;

For $1 \leq i \leq j \leq n$, let $B_{ij} = 1$

if $(\Pi(i), \Pi(j)) \in E$ and $B_{ij} = 0$

otherwise;

$\text{Commit}(B_{11}), \dots, \text{Commit}(B_{nn})$
 $\xrightarrow{\text{Commit}(\Pi)}$

$c \leftarrow \{0, 1\}$

\xleftarrow{c}

If $c = 0$, open all commitments

Else, open all commitments B_{ij}

where $(\Pi(i), \Pi(j))$ is in the
Hamiltonian cycle.

$\xrightarrow{\text{Openings}}$

Verify commitments
if $c = 0$, check that the
committed graph is
isomorphic to G
Else, check that a cycle
was opened.

Note that the simulator's first message Commit is computationally indistinguishable from the prover's first message, as the commitments are computationally hiding.

The simulator's second message is statistically indistinguishable from the prover's second message: if $c = 0$, the simulator does exactly what the prover does, and if $c = 1$, the simulator opens all commitments of some arbitrary cycle, which is the image of the Hamiltonian cycle under *some* permutation Π .

Finally, we need to argue that the **simulator is efficient**. Because the **commitments** in the first message are **computationally hiding**, the verifier cannot guess \hat{b} with non-negligible advantage, and thus $\Pr[\hat{b} \neq b] \leq 1/2 - \text{negl}(\lambda)$. \square

Knowledge Soundness & Proofs of Knowledge – Extractor

- If a prover often succeeds in convincing the verifier then it must know a witness
- PoK (existence of extractor) \Leftrightarrow Knowledge Soundness

Why do the simulator and extractor have special abilities?

Why do the simulator and extractor have special abilities?

- If simulator was succesful in polynomial time then simulator would be able to efficiently generate transcripts that are indistinguishable from real ones without any witness → impact on Soundess (a party with no witness could generate convincing transcripts)
- If extractor was succesful in polynomial time then a malicious verifier could potentially extract the witness too → witness leakage from provers messages (impact on Zero-Knowledge)

k-Special Soundness for 3 Round Protocols

- Given k accepting transcripts for the same instance x of L an extractor can compute a witness
- 2-Special soundness \rightarrow Classic definition of sigma protocols

Sigma Protocol

- 3 Round Protocol
- Perfect completeness
- Special soundness
- SHVZK

NIZK

- Under Random Oracle Model (but also more esoteric models like the Algebraic Group Model)
- Verifier doesn't need to interact with prover since it can locally draw challenge by querying the RO
- Zero-Knowledge: stems from RO programming to match the query of the Prover/Verifier
- NIZK-PoK Knowledge soundness stems from reprogramming the RO to get two different challenges
- Bonus Question: Suppose a Sigma protocol has soundness of $1/3$ what can you say about the soundness of its respective NIZK (after Fiat-Shamir)?

Thank you :-)

Any Questions?