

Zero Knowledge Proofs



<https://iuk.one/1033-1211>

Clemens H. Cap

ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock, Germany
clemens.cap@uni-rostock.de

Version 2



1. Conceptual Introduction
2. Commitment Schemes
3. Quisquater Metaphora
4. Chromatic Number of a Graph
5. Fiat-Shamir Identification
6. Feige-Fiat-Shamir Protocol

1. Conceptual Introduction

What is it about?

1. Conceptual Introduction

2. Commitment Schemes

3. Quisquater Metaphora

4. Chromatic Number of a Graph

5. Fiat-Shamir Identification

6. Feige-Fiat-Shamir Protocol

Basic Problem

Victor: Do you know the password?

Peggy: Yes.

Victor: Prove it.

Peggy: The password is "Black rose"

Problem 1: **Victor:** Thank you. Now I also know the password.

Problem 2: **Eve:** Shh... I am Eve. I was eavesdropping. and know the password.

1. Conceptual Introduction

More Elaboration (1)

Idea: Identity is given by an asymmetric (public, private) key pair.

Situation: Proving access rights by proving asymmetric identity

Peggy: "I know the private key belonging to this public key."

Victor: "Cool. Prove this!"

Attempt 1: Rendering the password

Victor: "Tell me the key!"

Peggy: "This defeats the purpose of asymmetric schemes."

Attempt 2: Signing messages as proof-of-password

Victor: "This is message X, please sign it"

Peggy: "X could be the SHA-256 hash of a document where I promise to pay 100 Bitcoin to Victor."

1. Conceptual Introduction

Discussion

Why do we sign hashes of documents instead of documents?

What, in a normal signing scenario, prevents the following attacks?

- 1 Victor presents a hash X which does not match the document to be signed.
- 2 Victor presents a hash X for which he has found a hash collision.

- Attempt 3:** Decrypting as proof-of-password
- Victor:** "This is one of your cipher texts C , please decrypt it"
- Peggy:** "Then you will learn the plain text of C "
"Then you will learn one pair of (cipher, plain) text"
"I protest against your adaptive chosen cipher text attack"

Formal Definition

A zero-knowledge interactive proof system for a password is a protocol where Victor and Peggy exchange messages about some password, at the end of which the following properties hold:

① Security:

- ① **Completeness:** If Peggy knows the password, Victor will be convinced of that.
- ② **Soundness:** If Peggy does not know the password, Victor will discover that.

② Zero Knowledge:

If Peggy adheres to the protocol and knows the password: Victor learns nothing beyond the fact that Peggy knows the password.

In particular: Victor learns nothing about a distribution on the set of possible passwords.

1. Conceptual Introduction

Remarks on the Definition

Note 1: Completeness and soundness hold **probabilistically**, ie. except for a very small probability, which can be pushed down with further rounds of the interactive protocol.

Note 2: If Peggy does not know the password, Victor may learn all kinds of things (such as particular attempts of Peggy to lie to him). However, this is not a problem.

Note 3: If Peggy does not adhere to the particular rules of the protocol, Victor may learn all kinds of things (including the password). However, this then is the fault of Peggy, not of the protocol.

2. Commitment Schemes

A new cryptographic tool required for Zero Knowledge Protocols

1. Conceptual Introduction
2. **Commitment Schemes**
3. Quisquater Metaphora
4. Chromatic Number of a Graph
5. Fiat-Shamir Identification
6. Feige-Fiat-Shamir Protocol

Cryptographic Anecdote: Stock Broker

Alice: I am a well known stockbroker.

Bob: Pick some good stocks for me. If they are winners, I will pay you a fee.

Alice: How do I know you will not invest in the stocks
and then forget to pay my fee?
I will rather tell you my selections for last month!

Bob: How will I know you are not selecting those stocks
of which you already know that they performed well last month?

Problem: Alice and Bob need a scheme where:

- Alice can commit to a choice she can not later change
- Bob cannot learn too early to what Alice committed

Cryptographic Anecdote: Who shall travel?

Situation:

- Alice is in the US, Bob is in China and they want to meet.
- Who shall travel?

Attempt 1:

- Alice suggests she throws a coin and if it is head then Bob has to travel.
- Alice says: "I threw a coin. It is head".
- Bob is not convinced and wants to participate in the decision.

Attempt 2:

- Alice suggests both throw a coin and if the result is equal then Bob has to travel.
- Bob says: "I threw a coin. It is tails"
- Alice replies: "Funny. I also got tails"
- Bob is not convinced.

Commitment Scheme

A commitment scheme is a protocol consisting of two phases:

Phase 1: Commit:

- 1 Alice chooses a value $v \in V$
- 2 Alice calculates a commit message c
- 3 Alice sends the commit message c to Bob

Phase 2: Reveal:

- 1 Alice tells Bob the value $v \in V$ she chose
- 2 Bob checks if this is compatible with the commit message c

Required Properties for Commitment Schemes

A commitment scheme must satisfy two properties:

Hiding Property: Bob cannot learn the value Alice has chosen from the commit message or anything about that value (eg. that one value is more probable than another value).

Binding Property: Alice cannot change the value she has chosen without Bob realizing this when comparing the commit message with the value Alice announced in the reveal phase.

Non-Solution: Hash Function

Procedure:

- 1 Alice and Bob agree on a cryptographically secure hash function
- 2 Alice chooses a value $v \in V$
- 3 Alice sends a hash $c := h(v)$ to Bob
- 4 Alice later reveals a value v' and Bob checks if $h(v') = c$
- 5 If $h(v') = c$, Bob concludes that $v = v'$

Problem: Violates the hiding property.

- 1 Bob iterates all possible values $x \in V$ and compares this to the hash value c .
- 2 Except in large sets V , Bob will learn the value too early.
- 3 Even in large sets V , Bob can probe for values, which tells him something about the probability distribution.

Non-Solution: Symmetric Encryption (1)

Procedure:

- 1 Alice and Bob agree on a symmetric algorithm
- 2 Alice chooses a secret key k
- 3 Alice sends $z := enc_k(v)$ to Bob
- 4 Alice later reveals v' and sends the key k to Bob
- 5 If $enc : k(v') = z$ Bob concludes that $v = v'$

Problem: Violates the binding property.

- 1 Alice can use a different key k' if she later wants to switch to a different value v'
- 2 Alice probes for keys until she finds a key k' such that $enc_{k'}(v') = z$
- 3 If the set V is small, this is very easy for her.

Non-Solution: Symmetric Encryption (2)

Further Elaboration:

- Assume a small set V .
- Objection: Will a different key k' decrypt z into an element of V ?
- Probably not. So where is the problem?
- This is true. But...
- Encryption algorithms usually come implemented with random padding.
- Why?
- If $V = \{0, 1\}$, then Alice has to match only the desired bit and claim the remainder to have been part of the random padding she had used (which Bob cannot check)

Solution 1: Symmetric Encryption with Random Padding

Procedure:

- 1 Alice and Bob agree on a symmetric algorithm to be used
- 2 Bob generates a random string R and sends it to Alice
- 3 Alice chooses a secret key k , a value v and sends $enc_k(Rv)$ to Bob
- 4 In the reveal phase, Alice sends Bob the value v and the key k
- 5 Bob decrypts the commit messages, verifies the string R and the value v

Binding Property: Due to the random string R , Alice cannot easily find a different key where the message decrypts to a different v but to the same R .

Hiding Property: Due to the encryption with an unknown key, Bob cannot learn the value of v

Solution 2: Hash Function with two step Random Padding

Procedure:

- 1 Alice and Bob agree on a hash function
- 2 Alice generates two random strings $R1$ and $R2$
- 3 Alice chooses a value v and sends $h(R1, R2, v)$ and $R1$ to Bob
- 4 In the reveal phase, Alice sends Bob all three items $(R1, R2, v)$
- 5 Bob compares this to the commit message

Binding Property: As soon as Bob knows $R1$, Alice can no longer change $R2$ or v without changing the value of the hash.

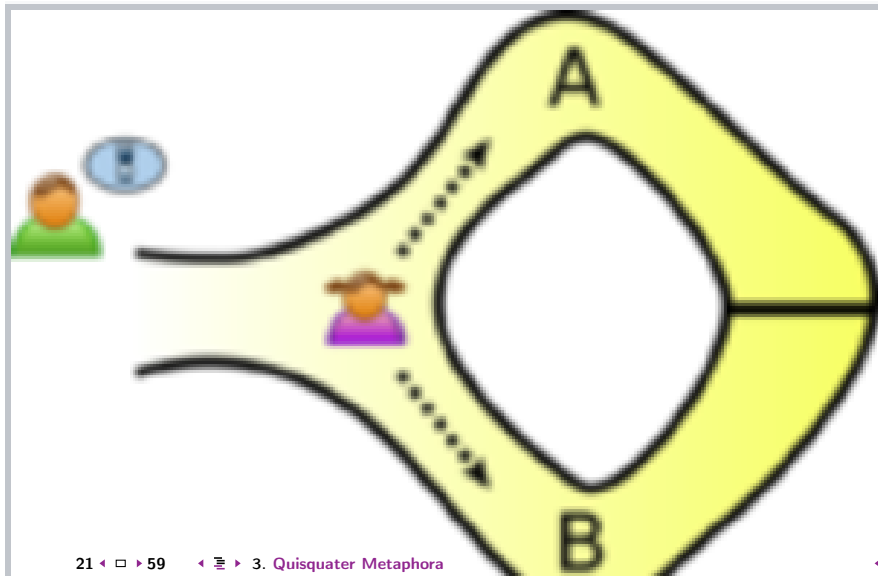
Hiding Property: Bob cannot run a brute force attack against v , since he does not know $R1$, which increases the size of his search space.

3. Quisquater Metaphora

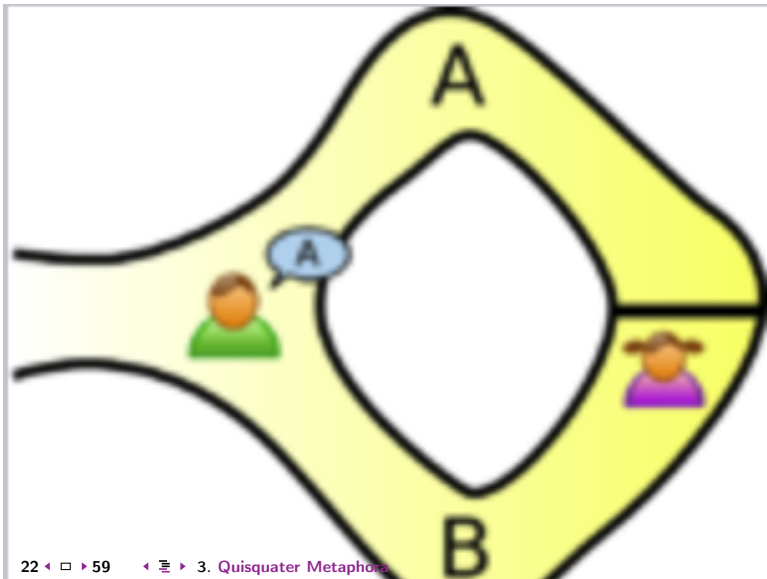
An easy way to understand Zero Knowledge Protocols

1. Conceptual Introduction
2. Commitment Schemes
3. Quisquater Metaphora
4. Chromatic Number of a Graph
5. Fiat-Shamir Identification
6. Feige-Fiat-Shamir Protocol

3. Quisquater Metaphora Setup

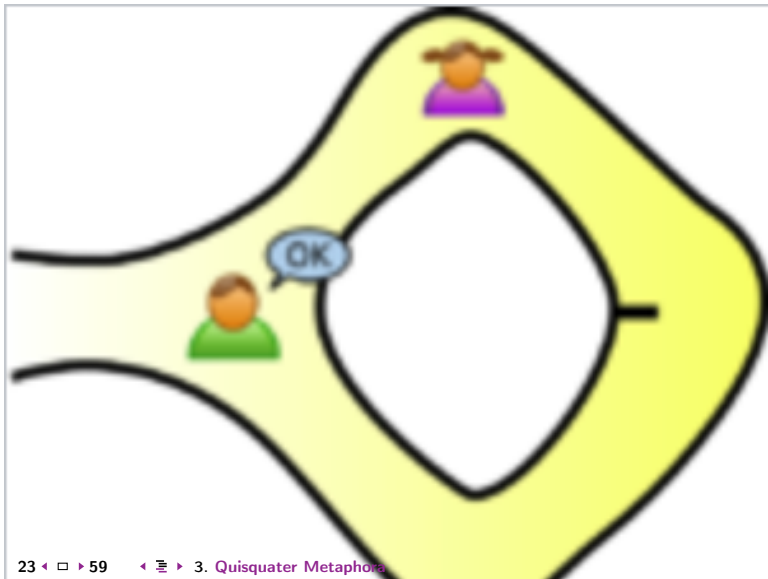


3. Quisquater Metaphora Problem



3. Quisquater Metaphora

Solution



4. Chromatic Number of a Graph

A first zero-knowledge protocol

1. Conceptual Introduction
2. Commitment Schemes
3. Quisquater Metaphora
- 4. Chromatic Number of a Graph**
5. Fiat-Shamir Identification
6. Feige-Fiat-Shamir Protocol

Preparation (1)

Concepts:

- ① Concept of a graph.
- ② Set of nodes N .
- ③ Edges as set of two-element node sets or as irreflexive binary relation on N .
- ④ Every node gets a color: $c : N \rightarrow C$.
- ⑤ Correct coloring: Nodes connected by an edge have a different color.
- ⑥ Chromatic number: Minimal number of colors needed for a correct coloring.

4. Chromatic Number of a Graph

Preparation (2)

Know: Let Γ be the class of all graphs γ which have a chromatic number of 3 or larger. Then the following problem is NP-complete:

Given $\gamma \in \Gamma$ in node-list form and an integer $k \geq 3$.

Does there exist a correct coloring of γ with k colors?

Observation 1: Checking a k -coloring is easy.

Observation 2: Deciding whether one exists (or finding one) is very complicated.

Know: The following problem is NP-complete:

Given a graph γ and a graph δ .

Does γ contain a subgraph which is isomorphic to δ is NP-complete.

Observation 1: Given a (subgraph) isomorphism it is easy to check that it is one.

Observation 2: Finding one is very complicated.

4. Chromatic Number of a Graph

Cryptographic Anecdote

Peggy: Knock, knock, admit me to the party

Victor: Ok. Today is Tuesday.
In my book it says I need to check that you can prove that the Tuesday-graph γ can be colored with 237 colors.

Peggy: This is fine. I can show you a coloring.

Victor: No. I am only the guard here. I am not allowed to know the solution.

Peggy: This is fine, I can give you a zero knowledge proof.

4. Chromatic Number of a Graph

Protocol (1)

Generation: Peggy generates an isomorphic graph γ' by producing a renumbering $r : \mathbb{N} \rightarrow \mathbb{N}$ as graph isomorphism and adds a few nodes (in ways which do not destroy the chromatic number)

Two boxes: Peggy prepares two boxes.

- 1 Box 1: Contains the isomorphism r
- 2 Box 2: Contains a coloring of the isomorphic (super)graph γ'

Commit: Peggy commits to the boxes using a commit scheme.

4. Chromatic Number of a Graph

Protocol (2)

Choice: Victor chooses one of the two boxes and asks Peggy to reveal the chosen box.

Case 1: It turns out to be box 1.

Victor checks that r is indeed an isomorphism.

This is at most quadratic in the number of nodes.

Case 2: It turns out to be box 2.

Victor checks that the coloring is indeed a true coloring.

This is linear in the number of edges.

Completeness and Soundness (1)

Observation 1:

- If Peggy knows a coloring, she can prepare boxes 1 and 2 for as many rounds as Victor wants to play.
- She can always generate a fresh isomorphism.
- She can use the original coloring to obtain a coloring of the isomorphic graph
- She can always add a few nodes without changing the chromatic number.

Observation 2:

- If Peggy can prepare boxes 1 and 2, she can invert the isomorphism and
- transport the coloring of the isomorphic graph back to the coloring of the original graph.
- leaving out the added nodes.

Observation 3:

- Peggy can always prepare one of the boxes in a way that it passes the verification.

Completeness and Soundness (2)

Observation 4:

- Peggy can prepare an arbitrary isomorphic graph and add a few innocent nodes.
- For this she does not need to know a coloring.

Observation 5:

- Peggy can always construct an arbitrary graph with a 217 coloring and make it look very complicated.
- For this she does not need to know a coloring of the original graph.
- Since subgraph isomorphism is NP-complete she will not be caught not using an isomorphic graph.

Observation 6:

- Peggy can pass with a chance of $1/2$ in a single round of the protocol, even if she does not know a coloring.

Zero Knowledge (2)

Computational versus statistical information:

- Since subgraph isomorphism is NP-complete, he cannot, computationally, obtain additional information from these graphs, helping him to solve the original problem.
- Statistically speaking: Victor learns something new.
- Computationally speaking: Victor cannot extract meaningful information from that.

Assume Victor is computationally unbounded:

- Victor can solve NP-complete problems.
- Then Victor can find the isomorphism and thus the coloring.
- But then Victor could also have solved the coloring problem himself.
- And, again, Victor would not have needed Peggy.

Notion of protocol simulation:

- Victor cannot simulate the protocol with Peggy completely.
- But Victor can simulate the protocol with Peggy in such a manner that a computationally bounded observer has no chance of finding out the difference between:

Case 1: Victor simulates Peggy.

Case 2: Peggy is not cheating and participating in the protocol.

Observations

Interaction:

- The mechanism crucially depends on a randomized interactive scenario.
- If Peggy knows in advance, which boxes Victor will open, she can cheat.

Nonces:

- The mechanism crucially depends on the use of nonces.
- Peggy must always use a different isomorphism.
- If Peggy uses the same isomorphism twice and Victor happens to open box 1 in the first and box 2 in the second attempt, he learns the secret.

Generalization:

- Every NP problem can be reduced to an NP-complete problem.
- For every question of the type $x \in \mathcal{L}$ and \mathcal{L} a language with an NP-hard recognition problem, a zero knowledge proof can be constructed.
- Still: The particular questions must be difficult for Victor

Abstract View

We use a set of problems which are

- complicated to solve
- whose solutions are easy to check
- which admit problem/solution isomorphisms which are complicated to determine

Public key of Peggy: An instance of the problem **Private key** of Peggy: A solution of this instance

4. Chromatic Number of a Graph

Protocol

Peggy transforms the problem into a random isomorphic instance of the problem
Peggy transforms the solution of her problem into a solution of the random instance

Peggy prepares

- Box1: The isomorphism (A proof that the isomorphic problem is isomorphic)
- Box2: The isomorphic problem and its solution

Victor learns isomorphisms

- which he can check whether they really are isomorphisms
- but which he learns nothing about the original solution

Victor learns solutions of isomorphic problems

- which he can check for solution
- but which do not tell him anything about the original solution
since he cannot figure out the isomorphism himself

5. Fiat-Shamir Identification

Number theoretic variants of
the graph theoretic protocol

1. Conceptual Introduction
2. Commitment Schemes
3. Quisquater Metaphora
4. Chromatic Number of a Graph
- 5. Fiat-Shamir Identification**
6. Feige-Fiat-Shamir Protocol

Fiat-Shamir Protocol

Motivation:

- Graph-based ZKP are burdensome for practical purposes.
- Large graphs do not fit on smart cards.
- A large graph as public key and a coloring as private key look odd.
- Both take much storage space.

Number-Theoretic Background

Let $n \in \mathbb{N}$ act as a modulus.

A number s is called a **square root modulo n** of the number t modulo n iff $s^2 \equiv t \pmod{n}$.

A **Blum integer** n is a product $n = p \cdot q$ of two prime numbers p, q with $p \equiv 3 \pmod{4}$ and $q \equiv 3 \pmod{4}$

Known: If a number is a square root modulo a Blum integer, then it has 4 different square roots.

Known: The following is equally computationally difficult:

- Factoring a Blum integer n .
- Finding the square roots mod n of a number k .

Assumed: Factoring large integers is difficult.

Setup of Fiat-Shamir Protocol

Modulus: Trusted third party generates and distributes large Blum integer.

Private key: Peggy chooses an s which is coprime to n as private key.

Public key: Peggy publishes $t := s^2$ as her public key.

Security:

- Computing square roots mod n is difficult.
- Calculating the private key s from the public key $t = s^2$ is difficult.
- Nobody can (easily) calculate the private key from the public key.

Fiat-Shamir Protocol (1)

Peggy chooses a random value r which is coprime to n .

Peggy computes $x = r^2 \bmod n$ and sends x to Victor.

Peggy prepares two boxes for Victor:

- Case 1: r
- Case 2: $rs \bmod n$

Victor picks one box and verifies the picked box:

- Case 1: $r^2 \equiv x \bmod n$
- Case 2: $(rs)^2 \equiv xt \bmod n$

Fiat-Shamir Protocol (2)

If the verification fails, Victor concludes that Peggy is a liar.

If the verification succeeds, Victor concludes that

- Peggy is telling the truth -OR-
- Peggy has succeeded in a 1-out-of-2 chance in lying to him.

Victor asks Peggy for more rounds in the protocol until he is satisfied with the remaining chance of Peggy lying.

5. Fiat-Shamir Identification Protocol is Secure (1)

If Peggy knows s , she can prepare the data in the two cases as required.

If Peggy does not know s , she will attempt to cheat.

Case 1:

- Suppose Peggy knew that Victor will chose case 1.
- Here it is easy to pass the test by following the protocol.
- In this case, Peggy would lie for the unchosen case 2, as she does not know s .

5. Fiat-Shamir Identification Protocol is Secure (2)

Case 2:

- Suppose Peggy knew that Victor will chose case 2.
- Peggy will prepare a random u and calculate $x \equiv u^2/t \pmod n$
- Division by t works, since t is coprime to n by construction.
- Peggy now sends (if asked for case 1) as x the value of u^2/t
- Peggy now sends (if asked for case 2) the value of u .

Victor will now check if $u^2 \equiv xt$.

Since Peggy chose x to be u^2/t Victor in fact checks if $u^2 \equiv (u^2/t) \cdot t$, which holds.

However, if Victor asked her unexpectedly for case 1, she cannot provide a reasonable r passing the test of $r^2 = x$ as she already has sent u^2/t for x and has no idea of the square root of t .

Protocol is Zero Knowledge

From case 1:

- When Victor receives information r he does not receive any information on s
- Obviously r is a random number.

From case 2:

- When Victor receives information rs he does not receive any information on s
- Multiplication by s is a bijection and does not change the random distribution.

What does Victor learn from x alone?

- x is not equi-distributed!
- After a while, Victor learns that x has the distribution of squares of random coprimes to n .
- This means, Peggy is following the (well-known) protocol.

If the distribution is different, then Peggy might be not following the protocol and no claim is made.

5. Fiat-Shamir Identification

Example

$$n = 3 \cdot 7$$

r	r^2	$rs(s=3)$	$rs(s=4)$	Remark					
0	0	0	0	Special	11	16	12	2	
1	1	3	4		12	18	15	6	Not coprime
2	4	6	8		13	1	18	10	
3	9	9	12	Not coprime	14	7	0	14	Not coprime
4	16	12	16		15	15	3	18	Not coprime
5	4	15	20		16	4	6	1	
6	15	18	3	Not coprime	17	16	9	5	
7	7	0	7	Not coprime	18	9	12	9	Not coprime
8	1	3	11		19	4	15	13	
9	18	6	15	Not coprime	20	1	18	17	
10	16	9	19						

Observations (1)

The statistics of column $r \cdot s$ look completely different if we have $s = 3$ or $s = 4$

The reason:

- We cannot divide by $s = 3$ (not coprime to modulus).
- We can divide by $s = 4$ (coprime to modulus).
- Thus: We really require s (equivalently: t) to be coprime to n
- Then we can divide and learn nothing.

Observations (2)

There are two values of squares which are coprime: 4, 16.

Each of it has 4 square roots.

There are four values of squares which are not coprime: 7, 15, 9 and 18.

They do not have 4 square roots.

There are the special cases 0 and 1.

Observations (3)

Root of very small values:

- For large n , the value 4 has a square root mod n of 2.
- It has another square root -2 , which is equal to $n - 2$.
- For small values of the private key (root)
- the public key is a normal (integer domain) square number
- the normal (integer) root algorithm quickly calculates a root (and the private key)

More precisely

- For a private key (root) k with $k \leq \sqrt{n}$
- the public key (square) is a normal (integer domain) square number.
- Then it is easy to calculate a square root mod n .
- Then it is easy to calculate the private key from the public key.
- Peggy must avoid choosing her public key as normal (integer domain) square.
- Thus, Peggy must avoid choosing the random s (private key) smaller than \sqrt{n} .
- Note: Above, with $n=21$, this did not work, with large n it does.

Observation (4)

Suppose Peggy chooses r in such a manner that Victor can determine a square root from $x = r^2$

Then Victor will choose case 2, learn $r \cdot s$, divide by r and learn s .

Thus: Peggy must avoid choosing too small random numbers.

Followup question: Does this restriction severely hamper Peggy's random choice of r ?

The random numbers she must avoid are in the interval 1 to \sqrt{n}

With \sqrt{n} bad apples in n possible choices, this leaves a portion of $1/\sqrt{n}$ bad apples.

With large n this does not pose a practical problem.

Importance of Using a Nonce

Suppose Peggy uses the same r in two rounds.

Victor will get a suspicion on this, if he sees the same x in two rounds.

If Victor observes this, he will ensure that he picks the other case in the second scenario.

From case 1 he learns r .

From case 2 he learns $r \cdot s$.

Dividing $r \cdot s$ by r he learns the secret key s .

This means, Peggy must store all previously used values of r

For practical purposes:

Peggy can ignore this and rely on the fact that a true random number generator will repeat values for large n only with an extremely small probability

5. Fiat-Shamir Identification

Reusing RSA modulus

Observation: In Fiat-Shamir, we can reuse the modulus.

Question: Can we reuse the modulus n in RSA?

Answer: No

If Bob knows the public key (e, n) and the private key d of an RSA scheme, then Bob can factor n into $n = p \cdot q$

If Alice uses a different RSA key pair with public key (e', n) and private key d' , with different e' but identical n , she can factor n into $p \cdot q$ Calculate Bobs private key d from Bobs public key e

6. Feige-Fiat-Shamir Protocol

Making Zero Knowledge Proof
non-interactive

1. Conceptual Introduction
2. Commitment Schemes
3. Quisquater Metaphora
4. Chromatic Number of a Graph
5. Fiat-Shamir Identification
6. Feige-Fiat-Shamir Protocol

Problems of the Fiat-Shamir Protocol

We need k rounds of communication to establish a security level of 2^{-k} .

Questions:

- What, if latency is too high for this?
- What, if we want to combine several rounds into one?
- Is there a non-interactive zero knowledge proof mechanism?

Answers:

- Specific situation: The Feige-Fiat-Shamir Protocol extends the Fiat-Shamir Protocol.
- General situation: There are two strategies for making the protocols non-interactive.

Feige-Fiat-Shamir Protocol

Setup:

- Let n be a Blum integer.
- Peggy chooses k coprime numbers s_1, \dots, s_k as private keys
- Peggy computes t_1, \dots, t_k with $t_j := s_j^2 \bmod n$ as public keys

Protocol:

- Peggy chooses a random r and sends $x := r^2 \bmod n$ to Victor.
- Victor chooses k bits $a_i \in \{0, 1\}$.
- Peggy computes $y := r \cdot s_1^{a_1} \cdot \dots \cdot s_k^{a_k} \bmod n$ and sends it to Victor.
- Victor checks that $y^2 = x \cdot t_1^{a_1} \cdot \dots \cdot t_k^{a_k} \bmod n$.

6. Feige-Fiat-Shamir Protocol

Analysis

Observations:

- Similar details as before – which we leave out here.
- We need only one round if we combine a sufficient number of keys.

Question

Question: Can every ZK protocol scheme be made non-interactive?

Interactive:

- Participants can communicate.
- Security derived from the number of rounds required by the verifier.
- Prover does not know the choice of the verifier.
- Prover cannot risk to cheat.

Non-Interactive Zero Knowledge Proofs

Parallel model:

- One exchange of data is sufficient.
- Roundtrip: Peggy to Victor, Victor to Peggy, Peggy to Victor.
- Combines an arbitrary number of rounds into one parallelized round.

Common reference string model:

- Peggy and Victor run a setup phase.
- During the setup phase both obtain access to the same (common) random reference string.
- Then Peggy sends a witness for the proof to Victor.
- Victor accepts or rejects.

Appendix

Contents of Appendix

List of Figures

LoF

Terms of Use

§

Citing This Document

→

List of Slides



List of Figures

1	21
2	22
3	23

Terms of Use (1)

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das **Zitatrecht** in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.

Alle sonstigen, hier nicht angeführten Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020. Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise **Par 60a UrhG** ermöglichen.

Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.

Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.

Use of Logos and Trademark Symbols: The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions <https://github.com/logos> to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status nicht oder nur mit unverhältnismäßig hohem Aufwand abzuklären ist. Ebenso kann den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen, obwohl deren Leistungen genutzt werden.

Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 ([Pressemitteilung](#), [Blog-Beitrag](#), [Urteilstext](#)). ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.

Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungs- und Anreizsystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungs- und Anreizsysteme digital erbrachter Leistungen haben.

Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.

Weitere Hinweise finden Sie im Netz [hier](#) und [hier](#) oder [hier](#).

Citing This Document

If you use contents from this document or want to cite it, please do so in the following manner:

Clemens H. Cap: Zero Knowledge Proofs. Electronic document. <https://iuk.one/1033-1211> 2. 7. 2023.

Bibtex Information: <https://iuk.one/1033-1211.bib>

```
@misc{doc:1033-1211,  
  author      = {Clemens H. Cap},  
  title       = {Zero Knowledge Proofs},  
  year        = {2023},  
  month       = {7},  
  howpublished = {Electronic document},  
  url         = {https://iuk.one/1033-1211}  
}
```

Typographic Information:

Typeset on ?today?

This is pdfTeX, Version 3.14159265-2.6-1.40.21 (TeX Live 2020) kpathsea version 6.3.2

This is pgf in version 3.1.5b

This is preamble-slides.tex myFormat©C.H.Cap

Title Page	1
Overview	2
1. Conceptual Introduction	
Basic Problem	4
More Elaboration (1)	5
Discussion	6
More Elaboration (2)	7
Formal Definition	8
Remarks on the Definition	9
2. Commitment Schemes	
Cryptographic Anecdote: Stock Broker	11
Cryptographic Anecdote: Who shall travel?	12
Commitment Scheme	13
Required Properties for Commitment Schemes	14
Non-Solution: Hash Function	15
Non-Solution: Symmetric Encryption (1)	16
Non-Solution: Symmetric Encryption (2)	17
Solution 1: Symmetric Encryption with Random Padding ...	18
Solution 2: Hash Function with two step Random Padding .	19
3. Quisquater Metaphora	
Setup	21
Problem	22
Solution	23

4. Chromatic Number of a Graph

Preparation (1)	25
Preparation (2)	26
Cryptographic Anecdote	27
Protocol (1)	28
Protocol (2)	29
Completeness and Soundness (1)	30
Completeness and Soundness (2)	31
Zero Knowledge (1)	32
Zero Knowledge (2)	33
Zero Knowledge (3)	34
Observations	35
Abstract View	36
Protocol	37

5. Fiat-Shamir Identification




Fiat-Shamir Protocol	39
Number-Theoretic Background	40
Setup of Fiat-Shamir Protocol	41
Fiat-Shamir Protocol (1)	42
Fiat-Shamir Protocol (2)	43
Protocol is Secure (1)	44
Protocol is Secure (2)	45
Protocol is Zero Knowledge	46
Example	47
Observations (1)	48
Observations (2)	49
Observations (3)	50
Observation (4)	51
Importance of Using a Nonce	52

Reusing RSA modulus	53
---------------------------	----

6. Feige-Fiat-Shamir Protocol

Problems of the Fiat-Shamir Protocol	55
Feige-Fiat-Shamir Protocol	56
Analysis	57
Question	58
Non-Interactive Zero Knowledge Proofs	59

Legend:

-  continuation slide
-  slide without title header
-  image slide