Zero-Knowledge Proofs Withstanding Quantum Attacks

T-79.515 Cryptography: Special Topics

Vesa Vaskelainen

Introduction

- *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks* (Ivan Damgård, Serge Fehr and Louis Salvail, Crypto 2004)

Introduction

- *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks* (Ivan Damgård, Serge Fehr and Louis Salvail, Crypto 2004)
- in a ZK proof of a statement, the verifier learns nothing beyond the validity of the statement

Introduction

- *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks* (Ivan Damgård, Serge Fehr and Louis Salvail, Crypto 2004)
- in a ZK proof of a statement, the verifier learns nothing beyond the validity of the statement
- it is natural to ask whether classical protocols are still secure if cheating players are allowed to run (polynomial time bounded) quantum computers?

Introduction

- *Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks* (Ivan Damgård, Serge Fehr and Louis Salvail, Crypto 2004)
- in a ZK proof of a statement, the verifier learns nothing beyond the validity of the statement
- it is natural to ask whether classical protocols are still secure if cheating players are allowed to run (polynomial time bounded) quantum computers?
- To study this question, two issues are important:
  1. The computational assumption on which the protocol is based must remain true even if the adversary is quantum
  2. More difficult question is whether the proof of security remains valid against a quantum adversary

- The first one rules out many assumptions such as hardness of factoring or extracting discrete logs

- The first one rules out many assumptions such as hardness of factoring or extracting discrete logs
- The major problem with the second issue is that in the classical definition of ZK, the simulator is allowed to *rewind* the verifier in order to generate a simulated transcript of the protocol execution.

- The first one rules out many assumptions such as hardness of factoring or extracting discrete logs
- The major problem with the second issue is that in the classical definition of ZK, the simulator is allowed to *rewind* the verifier in order to generate a simulated transcript of the protocol execution.
- If the prover and verifier are quantum, rewinding is not generally applicaple because when a quantum computer must produce a classical output, such as message to be sent, a measurement on its state must be done. State collapses and the original state cannot be reconstructed.

- The first one rules out many assumptions such as hardness of factoring or extracting discrete logs
- The major problem with the second issue is that in the classical definition of ZK, the simulator is allowed to *rewind* the verifier in order to generate a simulated transcript of the protocol execution.
- If the prover and verifier are quantum, rewinding is not generally applicaple because when a quantum computer must produce a classical output, such as message to be sent, a measurement on its state must be done. State collapses and the original state cannot be reconstructed.
- Thus, protocols that are proven ZK in the classical sense using rewinding of the verifier may not be secure with the respect to a quantum verifier. $\Rightarrow$ *Motivation of Damgård's, Fehr's and Salvail's work*

Results

- Three distinct techniques to convert an important class of classical honest-verifier ZK (HVZK) proofs into *quantum zero-knowledge* (QZK) proofs are proposed, these are the first practical ZK proofs withstanding active quantum attacks

Results

- Three distinct techniques to convert an important class of classical honest-verifier ZK (HVZK) proofs into *quantum zero-knowledge* (QZK) proofs are proposed, these are the first practical ZK proofs withstanding active quantum attacks
- The first technique assumes the existence of an unconditionally hiding trapdoor string commitment scheme and can be proven secure in the common-reference-string (CRS) model

Results

- Three distinct techniques to convert an important class of classical honest-verifier ZK (HVZK) proofs into *quantum zero-knowledge* (QZK) proofs are proposed, these are the first practical ZK proofs withstanding active quantum attacks
- The first technique assumes the existence of an unconditionally hiding trapdoor string commitment scheme and can be proven secure in the common-reference-string (CRS) model
- The second technique assumes the existence of any quantum one-way function and is secure in the CRS model

Results

- Three distinct techniques to convert an important class of classical honest-verifier ZK (HVZK) proofs into *quantum zero-knowledge* (QZK) proofs are proposed, these are the first practical ZK proofs withstanding active quantum attacks
- The first technique assumes the existence of an unconditionally hiding trapdoor string commitment scheme and can be proven secure in the common-reference-string (CRS) model
- The second technique assumes the existence of any quantum one-way function and is secure in the CRS model
- The third technique requires no computational assumptions and is provably secure in the plain model (no CRS)

Recap of Classical Protocols

- Let $R = \{(x, w)\}$ be a binary relation,
  $L_R = \{x \mid \exists w : (x, w) \in R\}$ the language defined by R.
  For $x \in L_R$, any w s.t. $(x, w) \in R$ is called a witness,
  $W_R(x) = \{w \mid (x, w) \in R\}$ the set of witnesses for $x \in L$.
- An (interactive) proof for a language $L = L_R$ is a protocol (P, V) between a probabilistic prover P and a probabilistic poly-time verifier V.

P		V
	common input x	
private $w \in W_R(x)$		
claims that $x \in L$		
	execution (P, V)	
		if $x \in L$ accept, $\Pr = 1$
		if $x \notin L$ accept, $\epsilon < 1$

- A Σ-protocol for a language $L = L_R$ is a three-move inteactive proof (P, V) for L

P		V
computes a 1st message a	$a \longrightarrow$	
	$\longleftarrow c$	chooses a random challenge c
computes an answer z	$z \longrightarrow$	decides accept/reject by applying a predicate $\text{Verify}_x(a, c, z)$

- special sound if the soundness-error ϵ equals the inverse of the number of possible challenges c
- i.e. if for $x \notin L$ any valid first message a uniquely defines a challenge c which allows an answer z with $\text{Verify}_x(a, c, z) =$accept

- An interactive proof (or argument) is called Zero-Knowledge (ZK) if for every poly-time verifier V there exist a poly-time simulator S, which takes as input $x \in L$ and outputs a simulated view of V in the execution of (P, V) on input x, indistinguishable from the real view.
- depending on the flavor of indistinguishability, ZK can be perfect, statical or computational
- Honest Verifier Zero-Knowledge (HVZK), means that it needs only be possible for a poly-time simulator to approximate the view of a verifier that follows the specified protocol

The Quantum Case

- ZK quantum interactive proof systems are defined as the natural generalization of their classical counterpart, letting prover be any quantum algorithm and verifier be any poly-time quantum algorithm
- Completeness, Soundness and the case when the proof is called an argument remains the same
- Quantum ZK (QZK) is defined as for the classical case except that the quantum simulator is required to produce a state that is exponentially close in the trace-norm sense to the verifier's view
- in the trace-norm sense is also defined perfect QZK, statistical QZK, and computational QZK

Classical Commitment Schemes

- classical (trapdoor) commitment schemes secure against quantum attacks do not require quantum computation, but they are guaranteed to remain secure even under quantum attacks.
- their construction is based on hard-to-decide languages with special-sound Σ-protocols and yields to the first unconditionally hiding string commitment schemes withstanding quantum attacks
- these commitments are used to construct QZK proofs
- A commitment scheme allows a party to commit to a secret s by publishing a commitment $C = \text{commit}_{pk}(s, \rho)$ for a random ρ s.t. the commitment C reveals nothing about s (hiding property) while on the other hand the committed party can open C to s by publishing (s, ρ) but only to s (binding property).

Alice	$\mathcal{G}(l) \Rightarrow pk$	Bob
has a secret s		
publishes a commitment		
$C = \text{commit}_{pk}(s, \rho)$	$C \longrightarrow$	
can open C to s		
by publishing (s, ρ)	$(s, \rho) \longrightarrow$	checks if $C = \text{commit}_{pk}(s, \rho)$

- $\neg\exists$ forger able to compute s, s' and ρ, ρ' s.t. $s \neq s'$ but $\text{commit}_{pk}(s, \rho) = \text{commit}_{pk}(s', \rho')$ (binding property)
- $\neg\exists$ distinguisher able to distinguish $C = \text{commit}_{pk}(s, \rho)$ from $C = \text{commit}_{pk}(s', \rho')$ with an advantage which cannot be ignored (hiding property)
- If the distinguisher (the forger) is restristed to be poly-time, the scheme is said to be computationally hiding (binding), while without restriction it is unconditionally hiding (binding)

Security in a Quantum Setting

- the computational or unconditional hiding property can be adapted in a straightforward manner by allowing the distinguisher to be quantum, the same holds for the unconditional binding property
- adapting the computational binding property in a similar manner results too weak definition
- in order to prove secure an application of a commitment scheme, which is done by showing that an attacker that breaks the application can be transformed in a black-box manner into a forger that violates the binding property, the attacker typically needs to be rewound, which cannot be justified in a quantum setting by the no-quantum-rewinding paradigm

Strong Enough Definition

- Their definition for the computational binding property of the commitment scheme is strong enough to prove QZK applications secure
- Idea of the definition is that it requires that it is infeasible to produce a list of commitments and then open (a subset of) them in a certain specified way with a probability significantly greater than expected.
- The definition uses a predicate Q, which models a condition that must be satisfied by the opened value in order for the opening to be useful for the committer.
- A commitment scheme $(\mathcal{G}, \text{commit})$ is called computational Q-binding if for every predicate Q, every polynomially bounded quantum forger $\mathcal{F}$ wins the game with probability $p_{\text{REAL}} = p_{\text{IDEAL}} + adv$, where adv is the advantage of $\mathcal{F}$, which is negative or negligible.

Trapdoor Commitment Scheme

- Besides the public-key pk, the generator $\mathcal{G}$ also outputs a trapdoor τ which allows to break either the hiding or the binding property.
- if the scheme is unconditionally binding, then τ allows to efficiently compute s from $C = \mathsf{commit}_{pk}(s, \rho)$
- if it is unconditionally hiding, then τ allows to efficiently compute commitments C and correctly open them to any s

A General Framework

- Assume a (statistical) HVZK special-sound Σ-protocol $\Pi = (a, c, z)$ for a language $L = L_R$, existence of an efficient generator $\mathcal{G}_{yes}$ generating $x \in L$ and $w \in W_R(x)$ and require that for every distinguisher $\mathcal{D}$ it is hard to distinguish a randomly generated yes-instance $x \in L$ from some no-instance $x \notin L$
- For such L, the following construction provides an unconditionally hiding and computationally Q-binding trapdoor string commitment scheme
- concrete languages which are believed to be hard to decide are proposed e.g. the Code-Equivalence (CE) problem, known to be at least as hard as the Graph-Isomorphism (GI) problem

$(\mathcal{G}, \textbf{commit})$

$\mathcal{G} = \mathcal{G}_{yes} \Rightarrow x \in L$ is parsed as pk, $w \in W_R(x)$ as τ

Alice		Bob
secret $s \in \mathcal{S} = \{0,1\}^t$		
commit$_{pk}$: generate (a, z, c)		
with HVZK simulator for Π,		
set $C = (a, s \oplus c)$	$C \longrightarrow$	(a, d)
open C to s	$s, c, z \rightarrow$	checks if $s \oplus c = d$ and
		Verify$_x(a, c, z)$ =accept

QZK proof protocol in the CRS model

- The common-reference-string (CRS) model assumes a string σ which is honestly generated according to some distribution and available to all from beginning.
- In the CRS model, an interactive proof is (Q)ZK if there exists a simulator which can simulate the (possibly dishonest) verifier's view of the protocol together with a CRS σ having correct joint distribution as in a real execution.
- The following shows how to convert any HVZK Σ-protocol into a quantum zero-knowledge (QZK) argument under the assumption that $(\mathcal{G}, \mathrm{commit})$ is an unconditionally hiding and computationally Q-binding trapdoor commitment scheme.

Let a HVZK Σ-protocol $\Pi =$(a,c,z). Assume w.l.o.g. that a and c sample first messages a and challenges c of fixed lengths r and t. Let an $(\mathcal{G}, \text{commit})$ with the domain $\mathcal{S} = \{0,1\}^{r+t}$ be given.

P	$\mathcal{G} \to pk$ CRS $= pk$	V
input x		input x
private $w \in W_R(x)$		
computes $a \leftarrow$ a		
chooses $c_P \leftarrow$ c		
commit$_{pk}(a \parallel c_P, \rho)$	$C \longrightarrow$	
	$\longleftarrow c_V$	chooses $c_V \leftarrow$ c
$z \leftarrow$ z$_x(a, c_P \oplus c_V)$	(a, c_P, ρ), $z \longrightarrow$	if $C =$ commit$_{pk}(a \parallel c_P, \rho)$ and Verify$_x(a, c_P \oplus c_V, z)$ =accept

Conclusions

- concrete QZK protocols which remain secure under quantum attacks and which do not need quantum computation or communication were obtained

Conclusions

- concrete QZK protocols which remain secure under quantum attacks and which do not need quantum computation or communication were obtained
- Does QZK proof systems exist without having to rely upon CRS?