On the power of quantum tamper-proof devices

J. Bouda
Faculty of Informatics - Masaryk University,
Botanická 68a, 602 00 Brno, Czech Republic,

P. Mateus, N. Paunkovic and J. Rasga
SQIG, Instituto de Telecomunicações - DM, IST, TULisbon
Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Abstract

We show how quantum tamper-proof devices (QTPD) can be used to attack and to develop security protocols. On one hand, we prove that it is possible to transfer proofs of zero-knowledge protocols using QTPD's. This attack can be extended to other security schemes where privacy is important. On the other hand, we present a fair contract signing protocol using QTPD's where there is no communication with Judge during the exchange phase (which is impossible classically). In the later case we make use of decoherence in the quantum state of the QTPD to implement a global clock over the asynchronous network. QTPD's seem to be possible to implement with existing quantum hardware, due to the fact that it is hard to isolate quantum memory from interference. These theoretical results contribute to justify the implementation of QTPD.

Keywords: Quantum tamper-proof device, zero-knowledge, contract signing.

1 Introduction

Recently, research on quantum hardware and software has been attracting a lot of attention, specially concerning applications in security and cryptography. On one hand, Shor showed that when a full fledged quantum computer is available all cryptosystems based on factorization and discrete logarithm will fail. On the other hand, quantum systems are more secure than classical ones due to the fact that it is not possible to *determine* a completely unknown quantum state, or in other words, to clone an unknown quantum state. Indeed, it is possible to establish a perfectly secure secret key over an (authenticated) quantum channel which can not be mimicked with classical channels.

In this paper we focus on the applications of quantum tamper-proof devices (QTPD). Indeed, quantum technology seems a very good candidate to develop tamper-proof devices due to its fragility and therefore, easiness to detect tampering. Moreover, quantum systems have inbuilt a pure random generator that

is very useful in security. Herein, we show how QTPD's can be used, on one hand, to attack existing secure systems, and on the other hand, to implement protocols that are not possible classically. In detail, we show how to attack with a QTPD the impossibility of transferring the proof for some class of zero-knowledge proof systems. Moreover, we present a fair contract signing protocol (which is impossible classically) using QTPD's.

Roughly speaking, a zero-knowledge proof system[1] is a two party protocol, between a verifier and a prover, in which the prover wants to show to the verifier that he knows some secret without conveying any information about this secret to the verifier (*zero-knowledge property*). Zero-knowledge proof systems are used in several protocols, like identification protocols[2], e-voting protocols[3], and e-money protocols[4]. While the zero-knowledge property is the most important security aspect these systems must fulfill, in the folklore another security property is assumed to hold in zero-knowledge systems. This property is known as *impossibility of transferring the proof* and it states that at the end of the protocol, the verifier cannot prove to a third party that he had interacted with the prover. So, the impossibility of transferring the proof is crucial in satisfying several anonymity properties for more complex protocols like undeniable signatures, identification protocols and electronic elections. In identification protocols the impossibility of transferring the proof allows the anonymity of the prover in the sense that the verifier can not show to a third party that a certain prover has identified to the verifier.

To prove the impossibility of transferring the proof in a particular zero-knowledge proof system it is usually assumed that the verifier has complete control over his memory space and that he can rollback to a previous state whenever he needs. In practice this constitutes a very generous assumption. In a more general context, the verifier may have restrictions accessing his memory, for instance, he might be able to read/write his memory only via a quantum tamper-proof device. In this case, we are able to set up an attack where the verifier transfers the proof with a QTPD to a third party.

Fortunately, QTPD's can also be used to create new protocols, and not only to attack existing ones. We will show how a QTPD with decoherence allows to set up a fair contract signing protocol.

Contract signing[5] is an important security task with many applications, namely to stock market and others. In a nutshell, a contract signing protocol is a two party protocol between Alice and Bob who share a contract and want to exchange each other's signature of the contract. However, Alice and Bob do not trust each other, and thus, the protocol must be *fair*. In other words, either both Alice and Bob get each other's signature or none of them does. A simple *fair* contract signing protocol is one where Alice and Bob do not communicate at all. Since none of them receives the signature, the protocol is fair. For this reason, it is also important to impose the protocol to be *viable*. In other words, if both parties behave honestly then both of them will get each other's signature.

It has been shown that there exists no fair and viable contract signing protocol[5], unless signing parties communicate with a trusted Judge in the signing phase. The proof is rather simple, and it is related with the impossibil-

ity of establishing distributed consensus in asynchronous networks[6]. Indeed, if such protocol exists it has a final step where one message is exchanged, say, from Alice to Bob. In this case Alice already has all information she requires to compute Bob's signature of the contract (and Bob does not). Therefore, if she does not send the last message, the protocol reaches an unfair state. One way to come around this difficulty is to relax the fairness condition probabilistically. Probabilistic fairness allows one agent to have ϵ more probability of binding the contract over the other agent. In this case, solutions where the number of exchanged messages between the agents is minimized have been found[7]. Another workaround is to consider optimistic protocols that do not require communication with Judge unless something wrong comes up[8]. The basic idea for this type of solution is to ask for the assistance of Judge when the contract signing protocol has initiated and some message is missing.

Herein we propose a completely new solution for fair contract signing. We assume a QTPD with decoherence that works as a tamper-proof global clock in the asynchronous network. In this way, Judge is able to know a posteriori whether the parties committed or not to the contract, upon a given time, without any communication (with Judge) *through the signing phase*.

The paper is organized as follows. In Section 2 we formalize QTPD's as quantum automata. In Section 3 we introduce the basic notion of zero-knowledge proof systems and the transference attack with QTPD's. In Section 4, we present the fair contract signing and finally, we draw some conclusions in Section 5.

2 Quantum Tamper-Proof Devices

In this section we formalize the concept of *quantum tamper-proof device*, that is, a machine for which an evil agent has no possibility of arbitrarily changing its internal state. We describe a quantum tamper-proof device as a quantum automaton. The notion of quantum automaton that we consider is a generalization of the notion introduced in [9], where we extend the set of possible outputs and allow measurements as inputs.

Definition 2.1 *A* quantum tamper-proof device *is a quantum automaton, that is, a tuple* $Q = (I, \Gamma, O, H, |\psi_0\rangle, U, A)$ *where:*

- I *is a finite set of* transition symbols*;*
- Γ *is a finite set of* observation symbols*;*
- $O \subseteq \mathbb{R}$ *is a finite set of* output symbols*;*
- H *is finite dimensional Hilbert space called the* state space*;*
- $|\psi_0\rangle \in H$ *is a unit vector in* H *called the* initial state*;*
- $U = \{U_i\}_{i \in I}$ *is a family of unitary transformations in* H *called* family of transitions*;*

- $A = \{A_\gamma\}_{\gamma\in\Gamma}$ *is a family of observables over* H *called* family of observables *such that the spectrum of* A_γ *is contained in* O *for all* $\gamma \in \Gamma$.

The dynamics of a quantum automaton is the following. Two types of inputs are possible: transition inputs I and observation inputs Γ. Transition inputs are associated to unitary transformations $\{U_i\}_{i\in I}$ while observation inputs are associated to observables $\{A_\gamma\}_{\gamma\in\Gamma}$. The automaton is a reactive machine, that changes state when an input is read and outputs a value when an observation input is read. The set of outputs O is a (finite) subset of the real numbers containing the spectrum of all observables A_γ of the automaton. The state of the automaton is a unit vector over the state space H, which is a space where all unitary transformations and observables are defined. The automaton starts at the initial state $|\psi_0\rangle$. The state of the automaton evolves deterministically as transition inputs are read and probabilistically when the inputs are observations, respecting the postulates of quantum mechanics. That is, when a transition input i is read, the state $|\psi\rangle$ of the automaton evolves to $U_i|\psi\rangle$ without outputting any value. If an observation input γ is read, the state $|\psi\rangle$ evolves to $P_o|\psi\rangle/\|P_o|\psi\rangle\|$ with probability $\|P_o|\psi\rangle\|$, for any eigenvalue $o \in O$ of A_γ, where P_o is the projection onto the eigenspace of A_γ associated to o. In this case, the automaton outputs o. As we shall see, we are usually not interested in unitary transitions, but only on observations.

To illustrate the concept we describe a quantum automaton that generates pure random bits.

Example 2.2 (Pure random bit generator) *Consider the quantum automaton* $(I, \Gamma, O, H, |\psi_0\rangle, U, A)$ *where:*

- $I = \{t\}$;
- $\Gamma = \{o\}$;
- $O = \{-1, 1\}$;
- $H = \mathbb{C}^2$;
- $|\psi_0\rangle = |0\rangle$;
- U_t *is the Hadamard transformation;*
- A_o *is the observable* $-|0\rangle\langle 0| + |1\rangle\langle 1|$.

Recall that the Hadamard transformation is the unitary transformation that maps $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Each time the sequence $t\,o$ of inputs is read the automaton generates a random bit. The bit generation works as follows. Initially, the input t associated with the Hadamard transformation is read, setting the state to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Then the observation input o is read, outputting -1 or 1 with probability $\frac{1}{2}$ each, and the state evolves to $|0\rangle$ or $|1\rangle$.

For obtaining more random bits, another sequence $t\,o$ should be read. Observe that for the next random bits, when t is read, the state evolves to $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. Nevertheless, when any of these states are observed by A_o the output is -1 or 1 with probability $\frac{1}{2}$.

We will also need to deal with quantum tamper-proof devices where the internal state is decaying over time for presenting the contract signing protocol. This feature is essential to model global clocks on the asynchronous network. For simplicity we will consider that the state is evolving over time until some measurement is performed. At that point the systems stops evolving and behaves as common quantum automaton (Definition 2.1).

Definition 2.3 *A* quantum tamper-proof device with decoherence *is a quantum automaton, that is, a tuple* $Q = (I, \Gamma, O, H, \rho, U, A)$ *where:*

- I *is a finite set of* transition symbols*;*
- Γ *is a finite set of* observation symbols*;*
- $O \subseteq \mathbb{R}$ *is a finite set of* output symbols*;*
- H *is finite dimensional Hilbert space called the* state space*;*
- ρ *is a map* $\rho : \mathbb{R}_0^+ \to DO(H)$ *describing the evolution of the initial state of the automaton if no input arrives, such that* $\rho(t)$ *is a density operator in* H*, that is* $\rho(t) \in DO(H)$*, and* $\rho(t)$ *evolves according to the quantum postulates;*
- $U = \{U_i\}_{i \in I}$ *is a family of unitary transformations in* H *called* family of transitions*;*
- $A = \{A_\gamma\}_{\gamma \in \Gamma}$ *is a family of observables over* H *called* family of observables *such that the spectrum of* A_γ *is contained in* O *for all* $\gamma \in \Gamma$*.*

The evolution of quantum tamper-proof device with decoherence is the following. The initial state is evolving according to $\rho(t)$, and after the first input arrives at time t_0, the quantum automaton with decoherence behaves exactly as a common quantum automaton without decoherence with initial state $\rho(t_0)$. From this point on, the evolution of the automaton is ruled only by the unitary transitions and observations.

3 Transferring Zero-Knowledge Proofs

One of the main motivations to build quantum computers is that they are able to attack widely used classical cryptosystems, like RSA and ElGammal. Besides involved quantum algorithms, it is also believed that simple quantum correlations may be used, in a distributed setting, to develop new attacks to security

protocols. In this section, it is shown how quantum tamper-proof devices, together with Bell pairs, can be used to transfer proofs of an important class of zero-knowledge proof systems.

In this section we show how quantum tamper-proof devices may be used to attack the impossibility of transferring the proof for the Goldreich, Micali and Wigderson[10] system. The importance of this system stands from the fact that all zero-knowledge proof systems can be reduced to ones similar to the Goldreich, Micali and Wigderson[10], and so the attack holds for all the protocols in this class.

A lot of research was pursued in understanding the power of zero-knowledge under computational restrictions, see for instance [11]. For the attack presented in this section it is assumed that all parties have the usual polynomial power (could be quantum or probabilisitic), and that the agents can use quantum tamper-proof devices.

A crucial feature of the attack is the use of tamper-proof Bell pairs. The tamper-proofing limits the verifier only to measure the first half of the Bell pair. Moreover, during the attack, the verifier can only perform two types of measurement over that half. These restrictions prevent the verifier to rollback the memory state of the Bell pairs after performing measurements, allowing a third party to check if the verifier was interacting with a particular prover, as shown below in the section.

Basic concepts

Zero-knowledge proof systems have became one of the most important concepts in computational security since their invention by Goldwasser, Micali and Rackoff in 1985 [1]. Before presenting zero-knowledge proof systems, we need to introduce several concepts from computational complexity. We assume that the reader knows the basics of computational complexity mainly the probabilistic oriented concepts (for a good reference see [12]).

To properly define an interactive proof system it is necessary to consider a model allowing agents restricted to probabilistic polynomial time computations. There has been some discussion about the properties this model should satisfy and consequently about its expressiveness and behavior [13, 14, 15, 16]. In this work, we introduce zero-knowledge proof systems in the same way as Goldwasser, Micali and Rackoff first introduced them in 1985 [1]. However, contrarily to what is common in this area, we will not make use of (interactive) Turing machines in proofs or when presenting algorithms since that level of detail does not bring new insights as far as this paper is concerned. Instead we will refer to the Church-Markov-Turing Postulate[1]. In the sequel we denote the set $\{0,1\}$ by 2, and given a word $x \in 2^*$, we denote the *number of bits* in x (also called *the length* of x) by $|x|$.

[1]The Church-Markov-Turing Postulate asserts that all the "reasonable" (classical) computational models are equivalent and inter-reducible in polynomial time (note that this reduction is not valid between classical and quantum models).

Definition 3.1 *An* interactive proof system (V, P) *is a protocol between two agents, the* verifier V *and the* prover P*, such that given an input* $x \in 2^*$*:*

- *Each agent has access to: (i) a private oracle that returns a (uniformly) random bit; (ii) the input* x *(only for reading); (iii) and a private memory.*
- *The protocol consists of several rounds and each agent is alternately active in each round. When active, each agent performs a computation and after that, either it sends a message to the other agent or the protocol halts.*
- *The computation of each agent depends only of the shared input* x *and of the messages received meanwhile from the other agent. The total time spent by the verifier* V *is polynomial in* $|x|$*.*
- *When the protocol halts,* V *outputs a value (denoted by* $(V, P)(x)$*) and, either accepts* x *(denoted by* $(V, P)(x)_{yes}$*), or rejects* x*.*

In the following, given an interactive proof system (V, P), the probability of the verifier to accept the input x is denoted by $\text{Prob}((V, P)(x)_{\text{yes}})$. Note that what makes the computation of the verifier to be probabilistic is the fact that the oracle returns random bits. Moreover, observe that in an interactive proof system only the verifier is restricted to do polynomial time computations. There is no such restriction for the prover. So, for instance, given any NP language it is not difficult to consider an interactive proof system in which the verifier accepts exactly the elements of that language, which we leave as an exercise to the reader.

Example 3.2 (Interactive proof system for SAT) *The verifier* V *and the prover* P *share a propositional formula* φ*. The protocol runs as follows:*

1. *If the formula* φ *is satisfiable, then* P *sends to* V *a valuation* v *that satisfies* φ *(note that the prover is not bounded to polynomial-time computations and therefore he can find this* v*). If the formula is not satisfiable, then* P *sends a random valuation to* V*.*
2. *When* V *receives a valuation* v*, it verifies whether the valuation satisfies* φ*. If this is the case it accepts* φ*, otherwise it rejects* φ*. In both cases, the output of the verifier, that is, the value* $(V, P)(\varphi)$*, is the valuation* v *received from the prover.*

Note that in Example 3.2, if $\varphi \in \text{SAT}$ then the verifier accepts φ with probability 1, and if $\varphi \notin \text{SAT}$ then the verifier accepts v with probability 0.

The formal definition of an interactive proof system is the following:

Definition 3.3 *A* binary language $L \subseteq 2^*$ has an interactive proof system *if there exists an interactive proof system* (V, P) *such that:*

1. *if* $x \in L$ *then* $Prob((V,P)(x)_{yes}) > \frac{2}{3}$;
2. *if* $x \notin L$ *then* $Prob((V,P')(x)_{yes}) < \frac{1}{3}$ *for any prover* P'.

The system (V,P) *is said to be an* interactive proof system for L.

One of the most significant results in the area of computational complexity concerning interactive proof systems was obtained by Shamir and states that PSPACE coincides with the class of languages for which there is an interactive proof system [17]. From a security point of view the concept of interactive proof system is much more useful when enriched with the notion of zero knowledge. Informally, an interactive proof system is zero knowledge if the verifier does not get any additional information by interacting with the prover. In the following, we will see how this additional property is important to obtain certain security goals. Prior to define what is zero knowledge we need to introduce the concept of *computational indistinguishability*.

Definition 3.4 *Two probabilistic algorithms* A_1 *and* A_2 *are* computationally indistinguishable *if, for any probabilistic polynomial-time decision algorithm* T*, polynomial* p *and large enough input* x

$$|Prob(T(A_1(x)) = 1) - Prob(T(A_2(x)) = 1)| \leq \frac{1}{p(|x|)}.$$

The notion of *computational indistinguishability* is important in cryptography to define several security concepts, as for instance, zero-knowledge proof systems.

Definition 3.5 A *binary language* $L \subseteq 2^*$ *has a zero-knowledge proof system* if L has an interactive proof system (V,P) such that, for any verifier V' there exists a probabilistic polynomial-time algorithm $S_{V'}$ such that, (V',P) is computationally indistinguishable from $S_{V'}$.

In what constitutes one of the fundamental results about zero-knowledge proof systems, in [10] it is shown that there is a zero-knowledge proof system for any language in NP provided that there is a non-uniform and indistinguishable cipher scheme. Zero-knowledge proof systems for languages in NP that are not believed to be in P can be employed to build useful security systems such as, identification protocols [2], e-voting protocols [3] and e-money protocols [4]. Informally theses systems allow the prover to show to the verifier that he knows a secret (a witness of an instance of the NP problem that is being considered) without revealing it. We now (re)define zero-knowledge proof systems from a cryptographic point of view.

Definition 3.6 *Let* L *be a language in NP that is believed not to be in P,* $x \in L$ *and* s *a witness*[2] *of* x*. Let* V *(verifier) and* P *(prover) be two agents.* P *states*

[2]Recall that a language $L \subseteq 2^*$ is in NP if there exists an polynomial-time decision algorithm A with two inputs, x and s, such that: (i) if $x \in L$ there exists a witness s such that $A(x,s) = 1$; (ii) if $x \notin L$ then for any witness s, $A(x,s) = 0$.

to know a witness s of x, and V wants to check whether P really knows s. A zero-knowledge proof system for x and s is an interactive proof system (V, P) such that the following conditions hold:

- Polynomial-time bound – *the overall time complexity of both V and P is probabilistic polynomial in the size in bits of x.*
- Soundness – *if P knows a witness s then V accepts that P has a witness s of x except with a negligible probability*[3]*;*
- Completeness – *if P does not know a witness s then V accepts that P has a witness s of x with negligible probability;*
- Zero-knowledge – *during the protocol V does not get any new information about the secret witness s of x.*
- Impossibility of transferring the proof – *the verifier V cannot show to a third party that he was interacting with P.*

Note that due to the polynomial-time bound condition in Definition 3.6 an agent in a zero-knowledge proof system can only exchange a polynomial (in $|x|$) number of messages, and moreover, the size of each message is also polynomially bounded over $|x|$. Furthermore, due to the zero-knowledge condition, a verifier does not obtain any new information about s, that is, it must be possible to simulate, in a computationally indistinguishable way, the interaction between the prover and an arbitrary verifier using a probabilistic polynomial-time simulator with input x.

It is worthwhile to briefly discuss and motivate each of the conditions presented in the Definition 3.6. The polynomial-time bound corresponds to the usual restriction in cryptography that all agents can only perform efficient computation. Soundness and completeness impose that the verifier should accept that the prover has the secret (except with a negligible error) if and only if the prover actually knows the secret (that is, a witness s of the instance x of the NP problem). The zero-knowledge property establishes that the verifier can not acquire additional information about the secret s during the run of the protocol. In particular, if the goal of the proof system is to identify the prover, i.e., the secret s identifies the prover, the zero-knowledge property guarantees that the verifier can not, after interacting with the prover, gain information to impersonate the prover. Finally, the impossibility of transferring the proof is an anonymity property. For instance, in identification protocols the impossibility of transferring the proof allows the anonymity of the prover in the sense that the verifier can not show to a third party that a certain prover has identified to him. We will see that this property can be attacked when tamper-proof (quantum) memory is available.

[3] A family of Bernoulli random variables $\{X_n\}_{n\in\mathbb{N}}$ is said to have negligible probability if, for any polynomial p and large enough natural number n, $\text{Prob}(X_n = 1) < \frac{1}{p(n)}$. In the context of soundness of zero-knowledge proof systems, the random variable X_n to consider takes value 1 if V does not accept that P has a witness s of x with size n.

One of the simplest examples of a zero-knowledge proof system according to Definition 3.6 is the Goldreich, Micali and Wigderson system [10] which is based on the conjecture that deciding if two graphs are isomorphic is not in P although it is in NP.

Example 3.7 (Goldreich, Micali and Wigderson system) *Suppose that both the verifier V and the prover P know two graphs G_0 and G_1 with n nodes. P says that he knows an isomorphism $\sigma : G_1 \to G_0$ (a witness that G_0 and G_1 are isomorphic). The protocol runs as follows:*

1. *P generates a random isomorphism $\pi : G_0 \to H$ and sends H to V;*
2. *V generates a random bit $b \in \{0, 1\}$ and sends it to P;*
3. *P sends the isomorphism $\tau = \pi \circ \sigma^b$ to V (where $\sigma^0 = id$);*
4. *V checks if $\tau(G_b) = H$.*

If the condition in Step 4 is not fulfilled, then V does not accept that P knows an isomorphism between G_0 and G_1, otherwise, Steps 1 to 4 are repeated n times, and the verifier only believes that P knows an isomorphism if in all these repetitions the condition in Step 4 holds.

It is interesting to observe that the problem chosen in Example 3.7, the graph isomorphism problem, is an NP problem that is not known to be NP-complete. In fact it would be expected that NP-complete problems were in general more secure than general NP problems since there is a polynomial time reduction from all NP problems to NP-complete problems. Nevertheless, NP-complete problems have not been widely employed in cryptography as opposed to cryptographic systems based on problems as the factorization and discrete logarithm (that belong to NP $\cap$ co-NP). The reason is that although NP-complete problems are very hard in the worst case, for some of them it is possible to find a witness for a random instance with high probability. Since the security of cryptographic systems rely on how hard it is to find those witnesses, NP problems believed to be sound to this attack, like graph isomorphism, are preferred.

Observe that the several steps of Example 3.7 constitute a typical iteration of a zero-knowledge proof system. The first step, designated by *commitment*, is when the prover sends the first message to the verifier. In the second step, called *challenge*, the verifier confronts the prover by typically sending him a bit. Finally, the third step, called the *decommitment*, is when the prover will have to send a message coherent with his commitment and with the challenge sent by the verifier. It can be shown that all the zero-knowledge proof systems for NP problems are reducible to a protocol with this structure [10]. The analysis and the attack presented in this Section rely on this structure of the protocol (commitment, challenge and decommitment). The results can be straightforwardly generalized to any system with this structure although they are presented with respect to Example 3.7. It is interesting also to note that graph isomorphism

seems not to be in BQP, that is, it seems to be hard to attack graph isomorphism using quantum computers with the techniques known nowadays [18].

The Goldreich, Micali and Wigderson zero-knowledge proof system described in Example 3.7 enjoys the following property, as shown in [10].

Theorem 3.8 (Goldreich, Micali and Wigderson) *The protocol of Example 3.7 is a zero-knowledge proof system.*

The technique employed to prove that the Goldreich, Micali and Wigderson system is a zero-knowledge interactive proof system for classical adversaries has become a widely used method to show the security of several protocols [19] and a well known technique to define security features [20, 13, 15, 16]. This technique is called *simulation paradigm*. This paradigm consists in showing that a real adversary of the protocol can be simulated by an ideal adversary, that is, an adversary that can not attack the system. For zero-knowledge proof systems the proof consists in showing that any dishonest verifier can be simulated by a process that does not interact with the prover (this process is the ideal verifier). Since the proof of Theorem 3.8 is important to understand the results of the paper we present it here.

Proof.[Theorem 3.8] Let V' be a verifier that might be dishonest, that is, a verifier that does not follow the protocol specified in Example 3.7. The goal is to show that it is possible to simulate the interaction between this verifier V' with an honest prover P, without communication with the prover. First, observe that the purpose of the simulator is to generate n tuples (H, c, τ) where H is randomly generated (since the prover is honest) and c is a bit generated accordingly by V'. Observe as well that the simulator has access to the code of V'. In these conditions, a verifier V' is a family of probabilistic polynomial-time algorithms $\{V'_k\}_{k\in 1\ldots n}$ where V'_k corresponds to the algorithm that V' uses to choose c in repetition k.

The algorithm V'_k receives a graph H and it may use some auxiliary data $w_k \in 2^*$ that he computed in previous repetitions. We assume that the initial data V' has about s is encoded in the string w_1. We now simulate V'_k as follows:

1. The simulator chooses $b \in \{0, 1\}$ randomly (with uniform distribution);
2. The simulator generates a random isomorphism $\tau : G_b \to H$ (with uniform distribution);
3. The simulator applies $V'_k(H, w_k)$ and verifies if the bit c computed by V'_k is b:
   (a) If $c = b$ then the simulation of repetition k is successful, since the tuple generated, (H, c, τ), has the same distribution as one created by interacting with the prover as discussed below. The auxiliary computation done by V'_k, which can be used in the next repetition, is stored in w_{k+1};

(b) If $c \neq b$ then the simulator jumps to Step 1.

Observe that the probability of $b = c$ is always $\frac{1}{2}$ (regardless of the computation of c by V'_k), since b was chosen uniformly. In other words, one expects that the simulator will have success in 2 trials. The simulator will not have success in n trials with probability $\frac{1}{2^n}$, which is negligible.

Finally, it is straightforward to show that the sequence of tuples (H, c, τ) generated by the simulator has precisely the same distribution of the tuples generated by an interaction between V' and P. For this reasons, these sequences are computationally indistinguishable.

The simulation paradigm technique ensures the impossibility of transference of proof, that is, the impossibility of tracing the interaction between the verifier and the prover. It is worthwhile to discuss this result since we will show how to attack it in a realistic context. The reason why the impossibility of transference of proof holds is the following. First, note that if the verifier wants to show to a third party, say Eve, that he has been interacting with the prover then he can send her the trace of the interaction, that is, the sequence of tuples (H, c, τ). Nevertheless, as shown in [10], regardless of the behavior of the verifier, it is always possible to generate, without any communication with the prover, a sequence of tuples computationally indistinguishable from the trace of the interaction. So, if Eve has only access to the trace of the interaction, she may not be persuaded that the verifier did in fact interact with the prover, because that sequence of tuples may have been generated by the verifier without communicating with the prover. For this point to hold the verifier must have complete control over the state of the computation. This scenario is too generous in what concerns the memory model of the verifier. If, at the beginning of the protocol, Eve gives a tamper-proof machine to the verifier such that he can not put the machine in an arbitrary state, then it is not possible to apply the simulation paradigm. It is also interesting to note that if quantum memory is shared by Eve and the verifier it seems hard to apply the simulation paradigm. The reason for this, is that if Eve and the verifier shared entangled memory and if measurements are performed by a dishonest verifier it seems hard to rollback. Therefore, the validity of the simulation paradigm in the quantum setting was an open problem. However, in [21], Watrous showed how to solve this problem. Again, in his proof Watrous assumes that the verifier has complete control over his private memory. As we will show in the next section in the context of the Goldreich, Micali and Wigderson system, if the memory of the verifier is tamper-proof, the verifier is able to show to Eve that he was interacting with the prover.

Quantum tamper-proof devices and transferring proofs

We are now able to define the specific type of quantum tamper-proof machine that will perform the attack to the Goldreich, Micali and Wigderson system. Before, we introduce the *computational observable* as $-|0\rangle\langle 0| + |1\rangle\langle 1|$ and the

diagonal observable as $-|\nearrow\rangle\langle\nearrow| + |\searrow\rangle\langle\searrow|$ where $|\nearrow\rangle$ is $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|\searrow\rangle$ is $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Definition 3.9 *A* quantum tamper-proof device for a graph G with n nodes *is the quantum automaton* $Q_G = (I, \Gamma, O, H, |\psi_0\rangle, U, A)$ *where:*

- $I = \emptyset$*;*
- $\Gamma = \{c, d\} \times \{1, 2\} \times \{1 \dots n\}^2$*;*
- $O = \{-1, 1\}$*;*
- $H = \otimes_{k=1}^{n} \otimes_{j=1}^{n} (\mathbb{C}^2 \otimes \mathbb{C}^2)$ *(space constituted by* n^2 *pairs of* qubits*);*
- $|\psi_0\rangle = \otimes_{k=1}^{n} \otimes_{j=1}^{n} (\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$*;*
- *The family of observables* A *is such that:*
  - $A_{(c,1,(k,j))}$ *corresponds to the computational observable over the first* qubit *of the* $(k \times j)$*-th pair with* $k, j \in 1 \dots n$*;*
  - $A_{(d,1,(k,j))}$ *corresponds to the diagonal observable over the first* qubit *of the* $(k \times j)$*-th pair with* $k, j \in 1 \dots n$*;*
  - $A_{(c,2,(k,j))}$ *corresponds to the computational observable over the second* qubit *of the* $(k \times j)$*-th pair with* $k, j \in 1 \dots n$*;*
  - $A_{(d,2,(k,j))}$ *corresponds to the diagonal observable over the second* qubit *of the* $(k \times j)$*-th pair with* $k, j \in 1 \dots n$*.*

  In general, $A_{(\underline{c},i,(k,j))}$ *denotes the* **c***omputational observable over the* i*-th* qubit *of the* $(k \times j)$*-pair while* $A_{(\underline{d},i,(k,j))}$ *denotes the* **d***iagonal observable over the same qubit.*

To perform the attack the verifier has to show to Eve that he interacted with the prover after the interaction ended[4].

In order to trace the interaction with a prover whose secret is an isomorphism between G_0 and G_1, Eve should set up a quantum tamper-proof device for G_0 with n nodes. This quantum automaton consists of n^2 pairs of qubits, all of them initially set at the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|\nearrow\nearrow\rangle + |\searrow\searrow\rangle)$. The automaton only accepts observation inputs, and the observables are exactly the computational and diagonal observables for each qubit (so in total there are $2 \times 2 \times n^2$ observables). We index the n^2 Bell pairs by (k, j) with $k, j \in 1 \dots n$. To understand the notation of observation inputs, the symbol $(c, 1, (k, j))$ is associated to the **c**omputational observable of the **1**st half of the (k, j) Bell

[4]Note that it is always assumed that the verifier can not communicate with Eve during the interaction with the prover. Otherwise Eve would play the role of the verifier, and therefore could check herself that the prover is interacting. The attack consists of the verifier holding an evidence that he interacted with the prover after this interaction ended.

pair, and similarly, $(d, 2, (k, j))$ is associated to the **d**iagonal observable of the **2**nd half of the (k, j) Bell pair.

We are now able to describe the protocol between Eve, the verifier and the honest prover, which will be used to obtain an evidence of the interaction between the verifier and the prover (this evidence is called a *trace of the interaction*). To this end, Eve and the verifier have to be colluded (nevertheless, notice that they do not trust each other otherwise Eve would always believe in the verifier), in the sense that Eve will give the quantum tamper-proof device to the verifier, which will be used to store the trace. The tamper-proofing consists on the verifier being restricted to apply only the observables $A_{(c,1,(k,j))}$ and $A_{(d,1,(k,j))}$.

Definition 3.10 Protocol to trace the Goldreich, Micali and Wigderson system *for a prover knowing an isomorphism between G_0 and G_1 with n nodes.*

1. *Eve and the verifier agree with a hash function*

$$h : \mathcal{G}_n \to \{c, d\}^n$$

   where $\mathcal{G}_n$ is the set of all graphs with n nodes.

2. *Eve gives the sealed automaton Q_{G_0} to V. The automaton is sealed such that V can only execute the observation operators $A_{(c,1,(k,j))}$ and $A_{(d,1,(k,j))}$.*

3. *V starts the interaction with P similarly to Example 3.7. For each repetition k:*

   (a) *In Step 1, V receives the graph H_k sent by P;*

   (b) *In Step 2, V computes $h(H_k) = e_{(k,1)} \dots e_{(k,n)}$ and applies the observables $A_{(e_{(k,j)},1,(k,j))}$ for all $j \in 1 \dots n$ obtaining the sequence of measurements $o_{(k,1)} \dots o_{(k,n)}$ with $o_{(k,j)} \in \{-1, 1\}$. Next, V computes $b_k = \oplus_{j=1}^{n}(1 + o_{(k,j)})/2$ and sends $b_k \in \{0, 1\}$ to P, where $\oplus$ denotes the xor operator.*

   (c) *In Step 3, V receives the isomorphism τ_k sent from P (and checks if $\tau_k(G_{b_k}) = H_k$).*

4. *When the protocol ends, V sends the machine to Eve together with the sequence $w = (H_1, b_1) \dots (H_n, b_n)$ of pairs (graph, bit) and the sequence $m = \tau_1 \dots \tau_n$ of isomorphisms obtained in each repetition of the protocol.*

5. *For each repetition $k = 1, \dots n$, Eve will do as following:*

   (a) *She starts by computing $h(H_k) = e_{(k,1)} \dots e_{(k,n)}$ and applies the observables $A_{(e_{(k,j)},1,(k,j))}$ and $A_{(e_{(k,j)},2,(k,j))}$ for all $j \in 1 \dots n$ obtaining the sequences $o_{(k,1)} \dots o_{(k,n)}$ and $r_{(k,1)} \dots r_{(k,n)}$, respectively, with $o_{(k,j)}, r_{(k,j)} \in \{-1, 1\}$.*

(b) *Then, Eve checks if* $o_{(k,j)} = r_{(k,j)}$ *for all* $j \in 1 \dots n$ *and if* b_k *(received from* V*) is equal to* $\oplus_{j=1}^{n}(1 + r_{(k,j)})/2$.

(c) *Finally, Eve checks whether* $\tau_k(G_{b_k}) = H_k$.

If for all repetitions the checks performed at Steps 5 b) and 5 c) hold then Eve believes that V *was interacting with* P.

The protocol above uses two features of quantum systems that assist to trace the interaction between the verifier and the prover.

The first feature is the fact that the outcome of a measurement of a Bell pair is a uniform Bernoulli random variable. Therefore, the measurement performed by the verifier in Step 3 b) will provide a Bernoulli random variable for b_k, that neither Eve nor the verifier will be able to predict. Note that if no randomness was employed, and the sequence $b_1 \dots b_n$ was predetermined by Eve, she could impersonate the prover while interacting with the verifier (jeopardizing completeness as presented in Definition 3.6). On the other hand, if the verifier has total control over the generation of the sequence $b_1 \dots b_n$, there is no way Eve can distinguish a simulation performed solely by the verifier with the interaction between the verifier and the prover (since it is always possible to create efficiently a simulation for any sequence of challenges $b_1 \dots b_n$). So, it is fundamental that the output of the machine is random, and that neither Eve nor the verifier have control over the sequence $b_1 \dots b_n$. In order for the verifier to be sure that the machine states are Bell pairs we can assume that a validity test is performed together with Eve when the quantum tamper-proof device is given to the verifier. At the light of this discussion, the QTPD must be tampered for both Eve and the verifier. This explains why the only way Eve observes her half of the qubits is through the inputs of the QTPD in step 4).

The second property of quantum systems that is used is entanglement. Bell pairs have the nice property that a measurement over one qubit will collapse both qubits to the same state. Since Eve does not trust the verifier, she uses Bell pairs to be sure that the measurements performed by the verifier in one half of the pairs are consistent with the measurements she performs in Step 5 a) over the second half. We can assume the second half of the qubits are always in possession of Eve, and that when the machine is given to the verifier it only contains the first half of the Bell pairs.

Finally, it is possible to establish that the attack is possible.

Theorem 3.11 *If the verifier can only change the state of the quantum tamper-proof device* Q_{G_0} *by executing* $A_{(c,1,(k,j))}$ *and* $A_{(d,1,(k,j))}$*, then the protocol presented in Definition 3.10 traces the interaction between the verifier and the prover for the Goldreich, Micali and Wigderson system.*

Proof. Assume that V wants to convince Eve that he was interacting with P without interacting with him.

By Step 4 of the protocol, V has to send to Eve a sequence $w = (H_1, b_1) \dots (H_n, b_n)$. We split the repetitions in two classes, those for which the verifier did not per-

form all measurements as described in Step 3b) and those for which he does all the measurements.

Suppose that at repetition k the verifier did not perform in Q_{G_0} at least one measurement $A_{(e_{(k,j)},1,(k,j))}$ (as indicated in Step 3b) for $h(H_k) = e_{(k,1)} \dots e_{(k,n)}$ and $k \in 1, \dots n$. Then, this measurement will be performed by Eve during Step 5a), and in that case, the probability of $\oplus_{j=1}^{n}(1 + o_{(k,j)})/2$ being equal to b_k is $\frac{1}{2}$.

Thus, if there are m repetitions for which at least a measurement $A_{(e_{(k,j)},1,(k,j))}$ was not performed, the probability of the verifier cheating Eve and not being detected is at most $\frac{1}{2^m}$.

Now, we deal with the remaining $n - m$ repetitions. We will split these repetitions in two, those in which the verifier did not perform the measurements accordingly to $h(H_k)$ and those where the measurements were made accordingly to $h(H_k)$. For the first case, it is straightforward to see that the probability of Eve detecting that V has cheated her, for each repetition, is $\frac{1}{2}$. Assuming that there are t of these repetitions, the probability of V not being detected is $\frac{1}{2^t}$. For the last case, we assume that there are $n - m - t$ repetitions. For each such repetition k, V obtains a sequence $o_{(k,1)} \dots o_{(k,n)}$. Since $o_{(k,j)} \in \{-1, 1\}$ is a uniform Bernoulli random variable and $b_k = \oplus_{j=1}^{n}(1 + o_{(k,j)})/2$, we have that $\text{Prob}(b_k = 1) = \frac{1}{2}$. Considering all the $n - m - t$ graphs over all repetitions, the number of graphs N for which the verifier will not be able to construct the isomorphism efficiently (we assume that this problem is hard) is a random variable with binomial distribution with parameters $\frac{1}{2}$ and $n - m - t$. For the verifier not to be detected while cheating, he has to change the N graphs in the trace to send to Eve. Since the probability of finding collisions for h is negligible, we assume that the image by h of each replaced graph differs at least one bit from the image by h of the original graph. In these conditions, the probability of Eve not detecting that V is substituting the N graphs in Step 5 is $\frac{1}{2^N}$.

Combining the results of all repetitions, the probability of V not being detected is given by $\frac{1}{2^{N+m+t}}$. The expected value of $N + m + t$ is minimum when $m = t = 0$ and takes the value $n/2$, leading to probability $\frac{1}{2^{n/2}}$ that is negligible in n.

4 Fair Contract Signing with QTPD's

In this paper we show that quantum memory with a particular type of decoherence enables fairness in contract signing. Indeed, our assumptions are close to those used by Crepeau [22] in order to achieve information secure quantum oblivious transfer. We only present a simple solution for the problem, that although fair and viable, has the disadvantage that an honest party who committed to the contract does not know whether the other party committed or not.

Quantum contract signing

The main idea of our contract signing protocol is to take advantage of decoherence in a tamper-proof device. In our particular case we consider the Werner states

$$\rho_{\eta(t)} = \eta(t)|\psi\rangle\langle\psi| + \frac{1-\eta(t)}{4}I, \tag{1}$$

decohering in time, where $|\psi\rangle$ is a maximally entangled state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and $\eta(t)$ is a function specifying the decoherence in time. In our particular case we use the function satisfying the property

$$|\eta(t) - 1| \leq \varepsilon \text{ for } t \leq t_0 \tag{2a}$$

$$|\eta(t) - 0| \leq \varepsilon \text{ for } t \geq t_1 \tag{2b}$$

for some fixed time points t_0 and t_1 (with $t_0 < t_1$) and a small nonnegative parameter ε. This time evolution describes a sudden decoherence given by the Werner state between time t_0 and t_1, and therefore, any measurement performed on the maximally entangled state before t_0 will lead to perfect correlations, whereas measurements made after t_c will be completely uncorrelated. It is assumed that no agent has control over the Werner state decoherence, that is, after time t_1 there is no entanglement between each part of the Werner state, whatever the agents do. Moreover, notice that the partial trace of the Werner state is stable over time (and it is $\frac{1}{2}I$).

Under this memory model assumption it is possible to construct a fair contract signing protocol requiring no communication between Judge and the signing parties during the exchange phase (which is impossible in the classical case). To this end, assume that both Alice and Bob each share with a trusted Judge $2n$ systems given by Werner states, as described in (1), and that completely decohere over the time period $[t_0, t_1]$. The objective is for Alice and Bob to commit with some contract at time t_0 and, upon committing at this time, it can not be possible for one agent to decommit from the contract after time t_1. If any agent commit (or decommit) during the period $[t_0, t_1]$ then there is no guarantee whether the contract is binding or not. It is important to stress that whatever an agent does, in any period of time, he cannot make the contract both binding and unbiding without negligible probability. The solution is given in Protocol 1.

During the certification procedure, if one agent, say Alice, wants to enforce the contract, she asks Judge to bind the protocol. Upon this request Judge compares the part of the Werner qubits shared with Alice and measured in the computational basis obtaining a sequence j_A. Then Judge asks Alice to show the result r_A she obtained when she measured her qubits. If r_A and j_A coincide, Judge gives the right to Alice to check on Bob commitment to the contract.

For the protocol not to be binding, Bob has to show that he did not committed to the contract. Bob can do this by showing to Judge the result r_B of measuring with the diagonal basis his part of the Werner states shared with Judge. Judge compares the outcomes of measurements performed on his systems

Protocol 1 Fair contract signing without bindingness verification

Participants: Alice, Bob, trusted Judge.
Input: times t_0 and t_1 (with $t_0 < t_1$), security parameter ε.
Goal: Alice and Bob want to sign a document simultaneously.
Requires: Alice and Bob share each with Judge $2n$ systems given by Werner states that decohere as specified by Eq. (1).

1: Alice signs the document and sends it to Bob (before time t_0).
2: Bob signs the document and sends it to Alice (before time t_0).
3: If Alice wants to commit to the contract, she measures her $2n$ qubits before time t_0 in the computational basis, otherwise she measures the $2n$ qubits in the diagonal basis.
4: If Bob wants to commit to the contract, he measures his $2n$ qubits before time t_0 in the computational basis, otherwise he measures the $2n$ qubits in the diagonal basis.
5: Judge chooses at random n of his qubits (shared with Alice) to measure in the computational basis and he measures the other n qubits (shared with Alice) in the diagonal basis. He measures his qubits whenever he wants. He does the same procedure for the $2n$ qubits shared with Bob.
6: If later Alice (or Bob) wants to enforce the contract, she sends the outcomes of her measurement to Judge who challenges Bob to state whether he committed or rejected the contract and to support his statement by the measurement outcomes.
7: Judge compares his measurements in the corresponding basis with Alice's and Bob's outcomes.
8: **if** all measurement outcomes are consistent **then**
9: If both decisions are *accept* then Judge accepts, otherwise he rejects.
10: **else if** measurement outcomes of either Alice or Bob are inconsistent **then**
11: Judge enforces decision of the consistent participant.
12: **else** ▷ *both Alice and Bob supplied inconsistent measurement outcomes*
13: In this case there is no honest participant to protect.
14: **end if**

in the diagonal basis obtaining j_B. If j_B and r_B coincide, then the contract is not biding, otherwise it is binding. Note that if Bob did measured his part of the Werner states with the computational basis (i.e, he committed to the contract) then he will not be able to produce the sequence j_B unless with exponentially low probability. The proof of this result follows:

Theorem 4.1 *(1) If both agents committed to a contract before time t_0 using n Werner states, then the probability of Judge unbinding the contract is exponentially low in the size of n, whatever the agents did before time t_1 (*binding soundness*). Similarly, (2) if one agent unbound the contract before time t_0 using n Werner states, then Judge will bind the contract after time t_1 with exponentially low probability in the size of n (*unbinding soundness*). Finally, (3) there is no strategy that an agent can perform that will make the contract both binding and unbiding without negligible probability on the size of n, after time t_1 (*completeness*).*

Proof. The binding soundness is easy. If both Alice and Bob committed to the contract before time t_0, that is, measured their part of systems in the computational basis, Judge's qubits and the agents' qubits are not entangled after time t_0. Moreover Judge qubits have collapsed to the computational basis. In this case, the probability that an agent is able to give the sequence of diagonal measurements for Judge's qubits is 2^{-n}. The unbinding soundness is similar.

Completeness follows by noticing that there is no entanglement between Judge system and the agent's system after time t_1. We will now show by reductio ad absurdum that there cannot be a strategy by Alice/Bob that will make the contract both binding and unbiding with non-negligible probability. The separation condition imposes that Alice/Bob cannot affect Judge system after t_1. This means that, if Alice/Bob can attack the protocol, they must have a prediction of Judge's measurements for both the diagonal and computational basis *before* time t_1. So assume that, say Alice, has a family of POVM's $E^n = \{E^i\}_{i \in M_n}$ (where $2n$ is the number of shared qubits with Judge and M_n is the index family denoting the outcomes of the POVM) that will allow her to generate two bit strings c and d of size n that will predict, before time t_1 and with *non-negligible probability in n*, the outputs of Judge's qubits in both the computational and diagonal basis. We will now show that such family of POVM's does not exist, even in the case of a joint POVM performed on all $2n$ qubits.

Assume that the way Judge chooses his measurement basis is uniform and independent of any computation that Alice does. This means that the best strategy that Alice can do if she predicts for qubit i, say $c_i = |0\rangle$ and $d_i| \nearrow\rangle$, is to, using a suitable POVM, try to prepare the i-th qubit of Judge to

$$\cos\left(\frac{\pi}{8}\right)|0\rangle + \sin\left(\frac{\pi}{8}\right)|1\rangle, \tag{3}$$

thus, maximizing the probability of success of achieving either 0 or + (this value is $\cos^2(\frac{\pi}{8})$). In this way, the probability of Alice being detected is $\sin^2(\frac{\pi}{8})$. This

value is a lower bound of the detection probability that the family of POVM's $E^n = \{E^i\}_{i \in M_n}$ can do, since it is not guaranteed that the POVM will be able to set Judge's qubit in (3) (for $c_i = |0\rangle$ and $d_i| \nearrow\rangle$). It is easy to see that for all possible combinations of c_i and d_i, the best strategy of Alice is to set Judge's qubit in some state similar to that in (3), which will lead to detection with probability $\sin^2(\frac{\pi}{8})$. Therefore, even in the best scenario, the probability of Judge not detecting that Alice is tricking is $\cos^{2n}(\frac{\pi}{8})$, which is negligible on n.

Noise and tamper-proof

Important is that the noise that transfers $\rho_{\eta(0)}$ to $\rho_{\eta(t_1)}$ (see Eq. (1)) affects the systems controlled by Alice and Bob, because otherwise all actions of Alice and Bob (namely the measurement) commute with the noise. On the other hand, it is sufficient if the noise affects only the systems controlled by Alice and Bob, that is the noise at Alice and Bob has to be tampered-proof. There is no need to require the decoherence on the quantum systems controlled by Judge. To be more formal, the quantum automaton with decoherence that we need for implementing the protocol is $Q = (I, \Gamma, O, H, \rho, U, A)$ where:

- $I = \emptyset$;
- $\Gamma = \{c, d\}$;
- $O = \{0, 1\}^{2n}$ is the set of outcomes of the measurements;
- $H = (\mathbb{C}^2 \otimes \mathbb{C}^2)^{2n}$ is the state space required for $2n$ maximally entagled pairs, where the first qubit is in the possession of Alice and the other with Judge;
- ρ is given by Eq (1);
- A_c corresponds to the observable associated to measuring all Alice's qubits in the computational basis;
- A_d corresponds to the observable associated to measuring all Alice's qubits in the diagonal basis.

We can achieve the same goal as in Protocol 1 using only single-qubit states. Judge can create $4n$ qubits randomly either in the standard or the dual basis state and send them to Alice and Bob. Afterwards - to check their commitment - he uses the knowledge of the distributed states instead of the measurement.

Analysis of the protocol

Unfortunately, Protocol 1 has one very unpleasant property. It is fair in the sense that none participant X can achieve the situation where he is sure the other participant Y bound the signature and at the same time Y is not sure whether X bound the protocol. Also, the protocol is correct in the sense that if both

participants want to sign the document and they behave honestly, the document is signed. It also true that in the contract exchange phase the Judge does not need to be contacted. However, without contacting Judge, the participant can never be sure whether the other agent's signature is binding or not, what is an unpleasant property in many situations. It is possible to improve this situation by including a protocol that increases the confidence of Alice and Bob on the commitment of each other to the contract. These improvements are being explored in [23].

On the other hand, we may easily find situations (cheating models) when Protocol 1 is perfectly sufficient. We specify them in terms of game theory and give some examples. Let S_A, S_B, R_A, R_B denote Alice and Bob's decision to sign/reject. We use the payoff functions to specify individual preferences of Alice and Bob:

	payoff of Alice		payoff of Bob	
	S_A	R_A	S_A	R_A
S_B	w_A	x_A	w_B	x_B
R_B	y_A	z_A	y_B	z_B

The values w, x, y, z denote respective payoffs of Alice and Bob. Our protocol is perfectly working as long as $w > x, y, z$ for both Alice and Bob, i.e. the most preferred situation for both participants is that the contract is signed. Such an example is e.g. an online airticket purchase - travel agent wants to be sure that he can cash the received electronic cheque and in case he cannot, the customer cannot use the ticket. The other way around, customer wants to be sure that he receives the ticket before the travel agent cashes the money. Most of them have no objectives into tricking the other party by rejecting the contract and making the other party believe that it is signed - this helps only to make financial loses to the travel agent and tricking the customer with invalid ticket.

5 Conclusions

The use of classical tamper-proof devices to attack classical protocols is further investigated in [24]. Observe that a quantum tamper-proof device can be almost simulated by a classical tamper-proof machine, but several problems arise. Randomness has to be replaced by pseudo-randomness, and moreover, decoherence has to be replaced by a global clock, which is not at all straightforward to achieve.

Concerning the attack to the impossibility of transferring proofs of the Goldreich, Micali and Wigderson system, it is more or less obvious to adapt the attack to zero-knowledge proof systems based on the commitment, challenge, response structure, which account for the majority of the zero-knowledge proof systems (an exception is non-interactive zero-knowledge systems that require a common random string). Attacks using only classical tamper-proof devices were developed in [24], namely attacks to undeniable and ring signatures.

Work is also being carried out on improving the contract singing protocol. The idea is to include a protocol that increases the confidence of Alice and Bob on the commitment of each other to the contract. These improvements are being explored in [23].

Acknowledgments

This work was partially supported by FCT (Portugal) and EU FEDER projects QSec PTDC/EIA/67661/2006 and project QuantLog FCT Project FEDER POCI/MAT/55796/2004. NP thanks the support from FCT (Portugal) and EU FEDER, grant SFRH/BPD/31807/2006. JB acknowledges support of grants GAČR 201/07/0603, MŠM0021622419 and GAČR 201/06/P338. We also thank the anonymous referees whose suggestions improve significantly the paper.

References

[1] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems. *SIAM Journal of Computing* **18** (1989) 186–208.

[2] A. Fiat and A. Shamir, How to prove yourself: practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, (Springer, 1987) pp. 186–194.

[3] R. Cramer, R. Gennaro, and B. Schoenmakers, A secure and optimally efficient multi-authority election scheme. In W. Fumy, editor, *EuroCrypt'97*, volume 1233 of *LNCS*, (Springer, 1997) pp. 103–118.

[4] D. Chaum, A. Fiat, and M. Naor, Untraceable electronic cash. In Shafi Goldwasser, editor, *Crypto'88*, volume 403 of *LNCS*, (Springer, 1990) pp. 319–327.

[5] S. Even and Y. Yacobi, Relations among public key signature systems. Technical Report 175, Technion, (1980).

[6] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson, Impossibility of distributed consensus with one faulty process. *Journal of ACM*, **32** (1985) 374–382,.

[7] Michael Ben-Or, Oded Goldreich, Silvio Micali, and Ronald L. Rivest, A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, **36** (1990) 40–46.

[8] N. Asokan, M. Schunter, and M. Waidner, Optimistic protocols for fair exchange. In *ACM Conference on Computer and Communications Security*, pp. 7–17, (1997).

[9] C. Moore and J. P. Crutchfield, Quantum automata and quantum grammars. *Theoretical Computer Science*, **206** (2000) 275–306.

[10] O. Goldreich, S. Micali, and A. Wigderson, Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, **38** (1991) 690–728.

[11] C. Dwork and L. Stockmeyer, Finite state verifiers i: the power of interaction. *Journal of ACM*, 39 (1992) 800–828,.

[12] C. H. Papadimitriou, *Complexity Theory*. Addison-Wesley, (1994).

[13] B. Pfitzmann and M. Waidner, Composition and integrity preservation of secure reactive systems. In *CCS'00*, pp. 245–254, (2000).

[14] R. Canetti, I. Damagrd, S. Dziembowski, Y. Ishai, and T. Malkin, On adaptive vs. non-adaptive security of multiparty protocols. In *EUROCRYPT'01* (Springer, 2001) pp. 262–279.

[15] J. Mitchell, A. Ramanathan, A. Scedrov, and V. Teague, A probabilistic polynomial-time calculus for analysis of cryptographic protocols. *Theoretical Computer Science*, (to appear).

[16] P. Mateus, J. Mitchell, and A. Scedrov, Composition of cryptographic protocols in a probabilistic polynomial-time process calculus. In R. Amadio and D. Lugiez, editors, *CONCUR'03*, volume 2761 of *LNCS*, (Springer-Verlag, 2003) pp. 327–349.

[17] A. Shamir, IP = PSPACE. *Journal of the ACM*, **39** (1992) 869–877.

[18] C. Moore, A. Russell, and J. Schulman. The symmetric group defies strong Fourier sampling. In *FOCS'05*, pp. 479–490, (2005)

[19] S. Goldwasser. Multi party computations: past and present. In *PODC'97*, pp. 1–6. ACM Press, (1997).

[20] R. Canetti. Security and composition of multi-party cryptographic protocols. *Journal of Cryptology*, **13** (2000) 143–202.

[21] J. Watrous. Zero-knowledge against quantum attacks. In *STOC'06*, pp 296–305. ACM Press, (2006).

[22] C. Crépeau. Quantum oblivious transfer. *J. Mod. Opt.*, **41** (1994) 2445–2454.

[23] J. Bouda, P. Mateus, and N. Paunkovic. Quantum contract signing. Technical report, SQIG-IT, (2008). Submitted for publication.

[24] P. Mateus and S. Vaudenay. Defeating privacy enhancement tools with seals. Technical report, SQIG-IT, (2008). Submitted for publication.