# Physical Cryptography

Mariana Costiuc[1], Diana Maimuţ[1], and George Teşeleanu[1,2]

[1] Advanced Technologies Institute
10 Dinu Vintilă, Bucharest, Romania
{mariana.safta,diana.maimut,tgeorge}@dcti.ro
[2] Simion Stoilow Institute of Mathematics of the Romanian Academy
21 Calea Grivitei, Bucharest, Romania

**Abstract.** We recall a series of physical cryptography solutions and provide the reader with relevant security analyses. We mostly turn our attention to describing attack scenarios against schemes solving Yao's millionaires' problem, protocols for comparing information without revealing it and public key cryptosystems based on physical properties of systems.

**Keywords:** Security education, recreational cryptography, physical cryptography, decoy based cryptography, Yao's millionaires' problem, public key cryptography.

## 1 Introduction

In our paper we present a security analysis to a series of problems that can be seen as abstract games. Our main motivation for studying such protocols is their teaching utility. Note that we are not aware of any real-world application of any sort, as these problems fall in the category of "recreational cryptography". Although recreational, these protocols can provide interesting insight and techniques that can be useful for understanding the concepts on which the underlying problems are based.

Physical cryptography [4, 11, 17, 20] makes use of physical properties of systems for encrypting and/or exchanging information (*i.e.* without using one-way functions). Although a very interesting teaching tool, it can be shown that some of the proposed methods are not safe in practice. Thus, our aim is to attack such physical protocols using methods similar to classical side channel techniques.

Besides the obvious cryptographic teaching utility of physical cryptography schemes, we believe that some of the schemes tackled in the current paper may be successfully used for introducing concepts corresponding to other domains. We provide the reader with such examples in the following sections.

Although some authors acknowledge that their proposed protocols are only useful for playing with children or introducing new concepts to non-technical audiences, the authors of [9–11, 21] claim that their schemes can be securely implemented in real-life scenarios. In [6], Courtois attacks one of the protocols proposed in [10], but the authors contest his results in [11]. We independently conducted a simulation of the attack and our results acknowledge Courtois' claim.

**Structure of the Paper.** In Section 2 we describe various schemes proposed in [9–11, 21] which aim at solving Yao's millionaires' problem and provide the reader with their corresponding security analyses. In Section 3 we present a set of protocols which act as solutions for comparing information without revealing it and discuss their security. In Section 4 we describe a public key cryptosystem constructed by means of an electrical scheme and tackle its security. We conclude in Section 5. Due to the page number restriction, we recall various physical cryptographic solutions which appeared in the literature in Appendix A. Also, in Appendix B we present a generic physical public key encryption scheme useful for introducing students to different properties of physical systems.

**Notations.** We denote by $U$ and $V$ the private spaces of Alice and, respectively, Bob. By "impenetrable" we further refer to an object that can not be broken or looked into no matter the means employed by an adversary. Note that, in practice, "impenetrable" objects do not exist, but we use this concept for presenting the philosophical aspects of different cryptographic problems.

## 2 Yao's Millionaires' Problem

In [24] Yao introduced "Two Millionaires' Problem". The problem can be defined as follows. Alice has a private number $a$ and Bob has a private number $b$. The goal of the two parties is solving the inequality $a \leqslant b$ without revealing the actual values. We further assume that $a, b \in [0, n]$ are integers.

In [9–11, 21] the authors present a number of solutions for the previously mentioned problem based on physical principles. In this section we focus on describing their proposed protocols together with our security analyses.

According to the original security model, during the following we consider Alice and Bob as being *honest but curious* users, *i.e.* they can observe, measure and compute whatever they like and try to get a hold on the other party's private numbers while following the protocol's steps.

### 2.1 "Elevator" Solution.

**Description.** To recall the scheme we follow the descriptions given in [9, 11]. We start by assuming that we have at our disposal a building with at least $n$ floors. Moreover, we consider that the chosen building is equipped with an elevator. Alice positions herself on floor number $a$ while Bob goes to floor number $b$. Then, Bob takes an elevator (from Bob's private space $V$) going down and stopping at every floor. Alice watches the elevator doors on her floor, making sure that Bob does not see her if the elevator doors open (here is Alice's private space $U$). If she sees the elevator doors open, she knows that Bob's number is larger. If not, then his number is smaller. Using such a protocol, Bob will not know the result of the comparison until Alice shares it with him.

**Security Analysis.** The only security considerations of [9, 11] are that Bob can lock the stairs and disable all elevators except one. This may prevent Alice from cheating by running between different floors to get a better estimate of Bob's number.

During our analysis we found other various attack scenarios. We consider the steps of the protocol as being sequential (*i.e.* first Alice gets to floor $a$ and then Bob gets to floor $b$).

1. If Alice uses the same elevator as Bob she can simply conceal a small camera[3] while ascending to floor $a$. Thus, she can recover $b$ as soon as Bob ascends to his designated floor. In order to mitigate such an attack, Bob must be ensured that Alice uses a different elevator or the stairs (*i.e.* making sure that Bob's elevator remains somewhat protected).
2. If the floor doors of Bob's elevator are not secured then Alice can open one of the doors and attach a motion sensor to the elevator. By analyzing the elevator's movement Alice can deduce $b$. Hence, Bob must be ensured that all the floor doors are secured against unauthorized access.
3. If Alice has access only to the stairs then she can install cameras on each of the $n$ floors[4]. If Bob limits Alice's access to only one floor ($a$) for security reasons, then he can always check the access readers installed on each floor and find $a$. These attacks can also be mounted by Alice if Bob takes the stairs. As a result, the only viable solution would be for Alice and Bob to use separate elevators.
4. Once Alice reaches $a$ then she can use a microphone to detect the sound made by the elevator's movement. By counting the number of times the elevator's engine starts or the doors open Alice can deduce $b$. Hence, to prevent such an attack, Bob can use a device for generating noise in order to mask the other relevant sounds. This attack can also be mounted by Bob for deducing $a$.

When Alice and Bob simultaneously ascend to their designated floors, the attack scenarios Items 3 and 4 are still feasible.

We do not claim that the protocol is feasible in practice (the doors must be "impenetrable" and the noise source must perfectly mask the sound of the elevator's movement). We only claim that the example can be practically used to introduce Yao's problem to non-specialized audiences and also to make people think of different methods of attacking the system.

### 2.2 "Race Track" Solution.

**Description.** For recalling the scheme we follow the description from [11]. Let us consider that Alice and Bob have at their disposal a race track of length $n$. Then, the two parties run toward each other from the opposite ends of the

[3] We can also consider all types of small devices which incorporate cameras.

[4] If the building already has security cameras, a simpler solution is bribing the security guard and watching the security footage to obtain $b$.

race track, maintaining the speeds of $a$ $m/s$ (Alice), respectively $b$ $m/s$ (Bob). The party which reaches first the midpoint of the track leaves a mark there and runs back, knowing that he/she was faster[5]. When the other party gets to the midpoint, he/she will know that he/she was slower[6]. In order to create their private spaces in this scenario, Alice and Bob have to construct an “impenetrable” fence across the track at the midpoint.

The authors of [11] state that the “race track” idea can be implemented on a computer if two different programs are allowed to work with the same file at the same time. Thus, consider that the shared file is a bit string of length $n$, with all bits initially equal to 1. Alice provides a program that goes over this bit string left to right, replacing the current 1 symbol by 0 at the speed of one symbol per $a$ time units. Bob provides a similar program going over the same bit string right to left, at the speed of one symbol per $b$ time units. When either of the two programs replaces $n/2$ symbols, it replaces the current symbol by $X$ and stops. In such a way, the two parties will know that whose program stops first has the bigger number. Both programs will have to use the computer’s internal clock.

**Security Analysis.** In [11] the authors mention that the “race track” solution only works if both parties are honest and provide the reader with an attack scenario otherwise. More precisely, the party who reaches the fence first does not run back but just waits to see when the other party arrives, thus figuring out the other party’s speed.

During our analysis we found that another restriction must hold. If Alice and Bob run on a circular track when they are “close enough”[7] to the midpoint they will be able to see each other. Thus, even if the parties are honest, the previous attack is still valid. To avoid such a scenario, a possible solution would be to put an “impenetrable”[8] fence such that both private spaces are isolated one from the other and also from the outside world[9].

The digital variant of the “race track” idea on a computer is, unfortunately, flawed. In order for the protocol to be valid both users need read/write access to the file. This implies that any of the parties can choose two positions of the other parties’ half of the file, continuously read the symbols corresponding to these positions and record the time needed for the symbols to change. This can be easily extended to monitoring multiple positions. Thus, each user can compute the other party’s value.

**Teaching Utility.** Although the digital variant is not secure, it can be used by teachers as an implementation task. Thus, students can implement two programs that race each other and also a third program that monitors the speed of either Alice and/or Bob.

[5] without knowing the actual speed of the other party

[6] again, without knowing the actual speed of the other party

[7] The precise difference between $a$ and $b$ depends on the race track’s radius.

[8] from both a visual and acoustic point of view

[9] If, for example, we isolate the two areas using only a wall, one of the parties can use a drone for spying the other.

### 2.3 "Communicating Vessels" Solution.

**Description.** To recall the scheme we follow the description from [11]. We start by assuming that Alice has a communicating vessel $C_A$ in her private space $U$, while Bob has a communicating vessel $C_B$ in his private space $V$. $C_A$ and $C_B$ are connected by a horizontal pipe attached to their bottoms and, thus, a working system is constructed. The shapes of the vessels are part of the parties' private keys. In the beginning the system is "almost" filled with water. Then, Alice starts pumping the water out of her vessel at the speed of $a$ gallons[10] per second, while Bob starts pumping the water in his vessel at the speed of $b$ gallons per second. The parties are simply watching whether the level of water is decreasing or increasing. If it is decreasing, then $a > b$; if it is increasing, then $a < b$.

**Security Analysis.** According to the authors of [21] the final level of water in the system depends not only on $a$ and $b$, but also on the shapes of both vessels. Also, the relation between $a$ and quantities that can be measured outside of Alice's vessel depends on the shape of Alice's vessel, which is unknown to anybody except Alice herself.

During our analysis we observed two main issues of the proposed protocol. First of all, if the participants pump water in and out of the system the shapes of their communicating vessels become irrelevant. In such a case, the authors might have thought about *pouring* water instead of pumping it while constructing their scheme. Secondly, the shapes of the vessels must be considered in such a way that the two parties can precisely measure fluctuations in their corresponding vessels. To explain this type of phenomena we can consider the following exaggerated example: the shapes of Alice and Bob's vessels correspond to those of two small artificial lakes and they pump water in and out with negligible speeds (*e.g.* a milliliter per hour). Then, they can not accurately detect which speed is greater than the other.

The scheme enhanced with our previous comments becomes equivalent with: Alice and Bob have two cylinder shaped vessels such that they can accurately measure fluctuations of the system. To detect Alice's value, Bob can use a graduated cylinder and measure the volume's fluctuation. Then, using his own speed value $b$ he can compute $a$. Hence, the scheme is insecure for solving Yao's problem but it can be used as a public key encryption scheme (see Appendix B).

**Teaching Utility.** Communicating vessels are a common example in physics teaching (see for example [12]). More precisely, the scheme provides a good opportunity for a teacher to introduce students to the dynamics of (ideal) fluids.

### 2.4 "Rope" Solution.

**Description.** For recalling the scheme we follow the description given in [10]. Alice and Bob privately select $c < 0$ and, respectively, $d > 0$. We position Alice and Bob in a plane, Alice at point $A = (a, c)$ and Bob at point $B = (b, d)$. Also,

[10] or whatever units

we give them both long pieces of rope. We assume that the scaling is such that Alice and Bob cannot see each other's point.

First, Alice fixes one end of her rope at point $A$ and selects as her private space $U$ a neighborhood of point $A$ that cannot be seen by Bob. Bob, too, selects $V$ as a neighborhood of his point $B$. Then, Alice fixes the other end of her rope to a random point $C$ in the plane, far enough so that her neighborhood $U$ can not be seen from $C$. After fixing the rope, she positions the part of the rope inside $U$ so that this part is not a straight line. She then communicates the coordinates of point $C$ to Bob.

Bob walks to point $C$, ties one end of his rope to Alice's rope, then walks back to his point $B$, while unwinding (not pulling) his rope along the way. When Bob reaches his $B$, he starts pulling the rope until Alice tells him to stop, which is as soon as Alice sees that the part of the rope inside her neighborhood $U$ is a straight line. To make sure that it is not by accident that the part of the rope inside her neighborhood $U$ is a straight line, Alice asks Bob whether or not the part of the rope inside his neighborhood $V$ is a straight line. If it is not, then Alice starts pulling her end of the rope toward her point $A$ until Bob tells her to stop, which is as soon as Bob sees that the part of the rope inside his neighborhood $V$ is a straight line.

When the parts of the rope inside both neighborhoods $U$ and $V$ are straight, Alice and Bob assume that their points $A$ and $B$ are connected by a straight rope, and they find the slope $s$ of the corresponding straight line by selecting any two points on the parts of the line inside their private neighborhoods. Then, $a < b$ if and only if $s > 0$.

**Security Analysis.** Some parts of the scheme described in [10] may seem redundant according to the authors. As pointed out by them, if both parties are honest the protocol can be simplified. To mitigate dishonest parties attacks, *e.g.* Alice must tell Bob to stop as soon as she sees that the part of the rope inside her neighborhood $U$ is a straight line. Otherwise, Bob could triangulate Alice's point $A$ by straightening the rope between $A$ and two different points of his choice.

Since we do not consider the honest but curious attack model for this precise protocol, another simple attack can be mounted. Bob can walk along Alice's rope until he is able to determine the coordinates of point $A$. To prevent Alice from seeing Bob while he tries to find $A$, he can use, for example, either a small drone or a powerful telescopic sight. To avoid such a vulnerability of the protocol, the neighborhood $U$ must be covered by an "impenetrable" material and, also, to contain a large number of points such that it is impossible for Bob to determine the exact position of $A$. When selecting the number of points in $U$ we also need to take into account the following scenario. After determining the precise position of $U$ in the plane Bob gets back to point $C$ and follows the initial protocol for determining $s$. Then, Bob can narrow down the number of possibilities for $A$.

**Teaching Utility.** A variation of this protocol for key exchange may be the following. Ted, a trusted third party, takes an infinite rope and fixes one end of it at Alice's point $A$. Similarly, Ted fixes another rope at Bob's point $B$. After fixing the ropes, Ted walks to a random point $T$ such that the distance to $A$ and $B$ is equal and then cuts the ropes at point $T$. In the last step of the protocol Ted returns the ropes to Alice and, respectively, Bob. The common key is the length of the two ropes.

Besides a good reason for a discussion about analytic geometry, this variations of the protocol can be the starting point for describing the secure key exchange protocol for the Internet of Things networks introduced in [18].

### 2.5 "Laboratory Scale" Solution.

**Description.** To recall the scheme we follow the description from [9]. We assume that Alice and Bob have access to a laboratory scale[11]. Each of the two parties manufacture a weight corresponding to their private number (*e.g.* in grams). We also assume that they have identical boxes[12] where each of them can put their corresponding weight. Alice enters the room where the scale is positioned and puts her box on one of the plates. Then, Bob enters and puts his box on the other plate. If his plate goes down, then his number is larger; otherwise, it is Alice's number that is larger.

**Security Analysis.** The authors argue in [9] that Alice and Bob do not have to be in the same place at the same time to perform the comparison, but they still have to be in the same place at some point, which may be inconvenient. In fact, if, say, Alice is worried about Bob cheating (by putting different weights on his plate to zoom in on Alice's weight), then she would have to stay in the room and watch what Bob is doing.

Note that when we analyzed the solution we assume that the box is "impenetrable". Compared to the "rope" solution where Bob needs to cheat in order to detect the dimensions of $U$, here Bob knows the precise size of the covering box. This gives him an upper limit of the weight's volume. If he knows the material of the weight, then he has an upper limit of the value $a$. This could be easily mitigated by keeping the weight's material secret.

## 3 Comparing Information Without Revealing It

The initial problem from which the study in [8] started is the following. Charlie complains to one of his managers, Alice, about a sensitive matter and asks her to keep it secret. A few months later, another manager, Bob, tells Alice that someone complained to him, also with a confidentiality request, about the same matter. Alice and Bob need a way to determine if the same person complained to them without revealing the identity of the complainer. The authors of [8] describe a series of complex protocols that try to accomplish this task. But, the

[11] a simple mechanism with two plates that are in balance when no weight is placed on either of them

[12] which, in this case, are considered their private spaces

simplest solution was actually provided by the 13 year old son of the first author: ”Why not just ask Charlie whether he complained to Bob?”. This proves that sometimes experts try to find too complicated solutions for simple things.

We further present a few solutions that can still work when implemented using our current technology. A legacy example may be considered the “airline reservation” solution. While Bob is not in the same room Alice calls a specific airline and makes a particular reservation in the name of her complainer. Then, Bob tries to cancel the reservation in his complainer’s name. Finally, Alice cancels or tries to cancel the reservation she made. It is obvious that nowadays such a version of the protocol can not be functional anymore, due to the fact that in order to cancel a reservation one needs to have extra pieces of information (*e.g.* the reservation code).

For uniformity, we consider, as in Section 2, that Alice and Bob are honest but curious.

### 3.1 Message for Bob

**Description.** We assume that Alice and Bob associate each candidate with a random telephone number. Alice dials the number[13] assigned to the person who complained to her (Charlie) and asks to leave a message for Bob. It is clear that the one answering the phone does not know who Bob is. A while after, Bob dials the number of the person who complained to him and asks if anyone has left him a message.

**Security Analysis.** The authors of [8] provide a short security analysis. More precisely: ① if Alice does not supervise Bob, then Bob might try several candidates and ② Dave might deny that a message was left for Bob.

The protocol was designed in a period of time in which telephones were only analog. But, nowadays, we also have digital and mobile phones. Thus, we further consider all the three cases when analyzing the security of the scheme. If Alice and Bob use the same phone to run the protocol, then, in the digital and mobile cases, Bob can check the call history of the phone to find out the identity of the complainer. Thus, to prevent such an attack, Alice must delete the call history. Even if she does this, there is a small probability that Dave will call back and, if Bob, is near the phone at that particular time, he can see the phone number and deduce the identity of the complainer. This problem can be easily rectified if Alice hides her number. Note that the previously mentioned problems do not happen in the analog case.

If Alice and Bob use different analog phones and Bob is nearby, he can redial the last number and ask Dave which is his phone number. Thus, in the analog case Alice needs to call another number afterwards[14]. In the digital case, Alice simply has to delete the call history to avoid the redialing attack. If the protocol is run using mobile phones, such an attack is even harder because Bob has to

[13] We denote the owner by Dave.

[14] to overwrite the call history

physically take Alice's phone. Even if he manages to snatch Alice's phone, the device might be locked.

We conclude that in the analog case either version is secure (*i.e.* with one or two phones) as long as Alice overwrites the call logs, while in the digital case it is better to use two phones. We believe that the protocol is secure as long as the initial scenario is valid[15] and our proposed countermeasures are taken into account.

### 3.2 Password

**Description.** We assume that Alice chooses to change her password in accordance with Charlie's name. Next, Bob tries to log in as Alice. In order to do so, Bob uses the name of the person who complained to him as a password.

**Security Analysis.** As in Section 3.1, Bob might try several candidates [8]. Additional to the initial security analysis, there is always the possibility that Alice installs either a key logger on the computer or a video camera inside the room and directly finds out Bob's password. Thus, the protocol is insecure.

**Teaching Utility.** In one version of the protocol, the authors of [8] suggest using the "passwd" Linux command to run the scheme. This provides a good opportunity for a teacher to introduce students to the Linux terminal basics and also how passwords are stored in Linux.

### 3.3 Cups

**Description.** We start by assuming that we have a small number $s$ of candidates. Alice and Bob get $s$ identical containers (*e.g.* by acquiring disposable cups), line them up and label them[16]. Then, Alice puts a folded slip of paper saying "yes" in the cup of Charlie and a slip saying "no" in the other $s-1$ cups. Bob does the same. Next, Alice and Bob remove the labels and shuffle the cups. To complete the protocol, both the parties look inside the cups to see whether one of them contains two slips saying "yes".

**Security Analysis.** If Alice and Bob use the suggested containers, Bob can always check which cup contains the slip saying "yes". Thus, it is better to use secure containers, for example ballot boxes which are tamper-evident. Hence, even if Bob manages to break into all the secure containers, Alice can detect that Bob cheated.

**Teaching Utility.** The secure version of the protocol may be seen as a toy version of the voting process. Thus, it can be used as an introduction to elections and electoral fraud.

[15] A powerful enough Bob can always eavesdrop the landline or ask the operator for Alice's call history.

[16] one for each candidate

# 4 Public Key Encryption

Several public key cryptosystems based on different laws of physics[17] can be found in [11][18]. Although these solutions are hard to implement in the real world[19], they provide a very good teaching tool. More precisely, a teacher can interactively transition from these toy protocols to precise explanations of the underlying physical laws.

Given the attack possibilities we observed while analyzing the schemes in [11], we chose to only discuss the "capacitors" solution during the following.

## 4.1 "Capacitors" Solution.

**Description.** Assume that Alice wishes to send a secret positive number $q_a$ to Bob. Let us consider that Alice has a capacitor $C_1$ of the capacitance $c_A$ (denoting her *public key*) and charge $q_A$ (denoting her *secret message*) in $U$. Similarly, Bob has a capacitor $C_2$ of the capacitance $c_B$ (denoting his *long-term private key*) and a randomly chosen charge $q_B$ (denoting his *session private key*) in $V$. Note that the private key is selected by Bob randomly before each transmission from Alice. The capacitors are connected in such a way that the plates holding the positive charges are connected by one wire, and the plates holding the negative charges are connected by another wire (see Figure 1). Alice has a switch that keeps the circuit disconnected until the actual transmission begins. Also, Alice has an ammeter to monitor the electric current in the circuit. Bob has a rheostat included in the circuit in $V$. This allows him to randomly change the resistance of the whole circuit, and therefore also to change parameters of the electric current during transmission.

According to the authors, Alice uses her switch to connect the circuit, starting the redistribution of the electric charges between the two capacitors. When this process is complete, she disconnects the circuit. After redistribution of charges, both Alice and Bob, have new charges: $Q_A$ and $Q_B$. Now, all that Bob has to do in order to compute the secret of Alice is to apply the following mathematical expression: $q_A = Q_B \cdot (1 + \frac{c_A}{c_B}) - q_B$.

**Security Analysis.** To promote an idea which might be relevant in practice, some experimental results should be presented. In this case, the authors gave an example of a system used for information transmission based on physical properties of passive components. Although the authors are theoretically right, Courtois contested the strength of their model in [6]. In our analysis, we propose a complete, yet simple way to demonstrate both theories. The proposed scheme is represented in Figure 2. In order to do so, we extended the electrical circuit proposed in [10] so that we could prove its functionality by simulating it. Based on the fact that the authors gave no technical specifications regarding the circuit, we analyzed several scenarios. The first one concerns the type of capacitors used

[17] We refer the reader to Appendix B.

[18] A similar solution for Yao's problem is described in [9].

[19] The authors assume that only Alice and Bob interferes with the system.

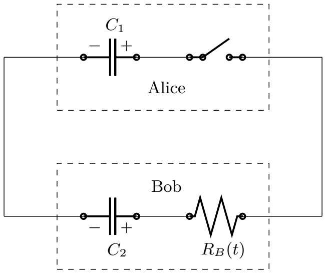

Fig. 1: "Capacitors" Solution

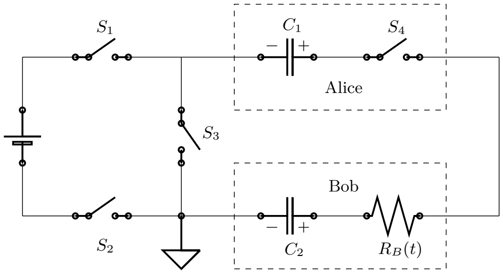

Fig. 2: Proposed "Capacitors" Solution

in the circuit. We tested the scheme using polarized and non-polarized capacitors with specific given input values and concluded that, in simulation, the differences are not significant. Nevertheless, in practice, the type of capacitor used is very important in order to avoid damaging the circuit.

To ease description, in order to validate the functionality of the "capacitors" solution we randomly choose a set of parameters for the scheme. Our example can directly be used in class to experimentally show that the solution is a viable one.

For obtaining a functional "capacitors" solution, we propose adding a power supply and 3 more switches (see Figure 2). The voltage generated by the power supply is 1 $V$. We use a 10 $\mu F$ capacitance for Alice's capacitor and a 1 $\mu F$ capacitance for Bob's capacitor. The rheostat is set at $R_1 = 431\ \Omega$ and $R_1 = 569\ \Omega$. The simulation is done using the electronic circuit simulator hosted by [1]. The first step of the simulation consists of charging the capacitors, in order to obtain the initial values for the electric charges. For charging the capacitors, switches $S_1$, $S_2$ and $S_4$ must be connected. After this step, the power supply is disconnected and the circuit is closed, meaning that switches $S_1$ and $S_2$ must be disconnected and switch $S_3$ must be connected. Switch $S_4$ is Alice's switch. Based on the values that were set as input, we measured the voltage drop $V_d$ on each capacitor and obtained the initial electric charges $q_A = 899.09\ nC$ ($V_{d_A} = 89.909\ mV$) and $q_B = 910.091\ nC$ ($V_{d_B} = 910.091\ mV$). After redistributing charges (*i.e.* when Alice connects the circuit) the charges become $Q_A = 10\ nC$ ($V_{d_A} = 1\ mV$) and $Q_B = 1\ nC$ ($V_{d_B} = 1\ mV$). In the final step of the protocol, Bob computes Alice's electric charge:

$$\begin{aligned} q_A &= Q_B \cdot (1 + \frac{c_A}{c_B}) - q_B \\ &= 10 \cdot 10^{-9} \cdot (1 + \frac{10 \cdot 10^{-6}}{10^{-6}}) - 910.091 \cdot 10^{-9} \\ &= -899.091 \cdot 10^{-9}\ C \end{aligned}$$

In [6], Courtois presents a rather intrusive attack in which Eve inserts a switch between Alice and Bob and measures the voltage (see Figure 3). In this case, switches $S_1$ and $S_2$ are disconnected. Switch $S_3$ is connected, Alice's switch

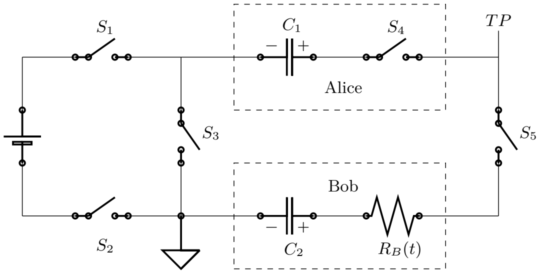

Fig. 3: Attack scenario “Capacitors” Solution

is $S_4$ and Eve’s switch is $S_5$. $S_4$ and $S_5$ are disconnected. Eve measures the voltage between Alice and Bob, right after Alice connects her switch. After the measurement, Eve connects her switch too. This is a very simple way to determine $V_{d_A}$. Since Alice’s capacitance is a public parameter, Eve just computes:

$$\begin{aligned} q_A &= c_A \cdot V_{d_A} \\ &= 10 \cdot 10^{-6} \cdot 89.909 \cdot 10^{-3} \\ &= 899.09 \cdot 10^{-9}\ C \end{aligned}$$

After running the simulation, we observed that the attack scenario is a plausible one. Note that the detection of Eve’s attack depends on the quality of the equipment that she possesses.

Initially, for protecting the circuit we thought of adding a plus of security by connecting each capacitor to a different power supply. It turned out this is not enough, since Eve can measure the circuit in any point which surrounds each Alice’s and Bob’s private space. Thus, we dropped the idea and choose the simpler version of the two.

## 5 Conclusions

We recalled various physical cryptographic solutions and discussed their security in the “honest but curious” model. Thus, we provided the reader with different attacks scenarios against a set of schemes solving Yao’s millionaires’ problem, a number of protocols for comparing information without revealing it as well as a a public key cryptosystem based on physical properties of systems.

## 6 Acknowledgments

The authors would like to thank Valentin Petre for his helpful comments on the “Communicating Vessels” solution.

## References

1. Falstad Electronic Circuit. https://www.falstad.com
2. The Diffie-Hellman Key Exchange Using Paint. https://www.youtube.com/watch?v=3QnD2c4Xovk
3. Balogh, J., Csirik, J.A., Ishai, Y., Kushilevitz, E.: Private Computation Using a PEZ Dispenser. Theoretical Computer Science **306**(1-3), 69–84 (2003)
4. Bell, T., Thimbleby, H., Fellows, M., Witten, I., Koblitz, N., Powell, M.: Explaining Cryptographic Systems. Computers & Education **40**(3), 199–215 (2003)
5. Bultel, X., Dreier, J., Lafourcade, P., More, M.: How to explain modern security concepts to your children. Cryptologia **41**(5), 422–447 (2017)
6. Courtois, N.T.: Cryptanalysis of Grigoriev-Shpilrain Physical Asymmetric Scheme With Capacitors. IACR Cryptology ePrint Archive (2013), http://eprint.iacr.org/2013/302
7. Crowley, P.: Mirdek: A Card Cipher Inspired by ”Solitaire”. http://www.ciphergoth.org/crypto/mirdek/
8. Fagin, R., Naor, M., Winkler, P.: Comparing Information Without Leaking It. Communications of the ACM **39**(5), 77–85 (1996)
9. Grigoriev, D., Kish, L.B., Shpilrain, V.: Yao's Millionaires' Problem and Public-Key Encryption Without Computational Assumptions. Int. J. Found. Comput. Sci. **28**(4), 379–390 (2017)
10. Grigoriev, D., Shpilrain, V.: Secure Information Transmission Based on Physical Principles. In: UCNC 2013. Lecture Notes in Computer Science, vol. 7956, pp. 113–124. Springer (2013)
11. Grigoriev, D., Shpilrain, V.: Yao's Millionaires' Problem and Decoy-Based Public Key Encryption by Classical Physics. Int. J. Found. Comput. Sci. **25**(4), 409–418 (2014)
12. Halliday, D., Resnick, R., Walker, J.: Fundamentals of Physics. John Wiley & Sons (2010)
13. Khovanova, T.: One-Way Functions. https://blog.tanyakhovanova.com/2010/11/one-way-functions/
14. Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC press (1996)
15. Moran, T., Naor, M.: Polling with Physical Envelopes: A rigorous Analysis of a Human-Centric Protocol. In: EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 88–108. Springer (2006)
16. Moran, T., Naor, M.: Basing Cryptographic Protocols on Tamper-Evident Seals. Theoretical Computer Science **411**(10), 1283–1310 (2010)
17. Naor, M., Naor, Y., Reingold, O.: Applied Kid Cryptography or How to Convince Your Children You Are Not Cheating. http://www.wisdom.weizmann.ac.il/~naor/PAPERS/waldo.pdf
18. Nishigami, K., Iwamura, K.: Geometric pairwise key-sharing scheme. In: SecITC 2018. Lecture Notes in Computer Science, vol. 11359, pp. 518–528. Springer (2018)
19. Quisquater, J.J., Quisquater, M., Quisquater, M., Quisquater, M., Guillou, L., Guillou, M.A., Guillou, G., Guillou, A., Guillou, G., Guillou, S.: How to Explain Zero-Knowledge Protocols to Your Children. In: CRYPTO 1989. Lecture Notes in Computer Science, vol. 435, pp. 628–631. Springer (1990)
20. Schneier, B.: The Solitaire Encryption Algorithm. https://www.schneier.com/academic/solitaire/

21. Shpilrain, V.: Decoy-Based Information Security. Groups Complexity Cryptology **6**(2), 149–155 (2014)
22. Singh, S.: The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. Anchor (2000)
23. Stephenson, N.: Cryptonomicon. Arrow (2000)
24. Yao, A.C.: Protocols for Secure Computations. In: SFCS'82. pp. 160–164. IEEE Computer Society (1982)

## A Recreational Cryptographic Problems

The interest of the cryptographic community regarding various recreational cryptography problems has grown in time. We further recall a series of physical cryptographic solutions which appeared in the literature. Note that our list of recreational cryptographic problems is, by no means, extensive.

**"Finding Waldo" Solution.** The authors of [17] provide an insight on how to convince people about knowing Waldo's location without revealing it. We initially assume that Alice and Bob have a large piece of cardboard[20]. As a first step, Alice cuts a Waldo shaped hole in the middle of the cardboard. To prove that she knows where Waldo is, Alice puts the shape precisely on top of Waldo while Bob is not looking and then calls Bob to check. Given the previous steps of the protocol, Bob learns nothing about the location of Waldo. Next, Alice must prove that she has the correct Waldo picture. Therefore, she must pull the book beneath the cardboard in front of Bob's eyes without revealing information about the place from which she is pulling the book[21].

**"Ali Baba Cave" Solution.** A well known story for explaining the intuition behind zero knowledge protocols is presented in [19]. The story is about a magical cave shaped like a ring with an entrance on one side as well as a magical door blocking the opposite side. We assume that Alice discovers the secret magical word that opens the door and wants to prove to Bob that she knows the secret without revealing it. Thus, they agree to label the left and right paths from the entrance `head` and `tail`. The protocol proceeds as follows. Bob waits outside the cave as Alice goes in. Then, Alice flips a coin to determine the path she follows. Note that Bob is not allowed to see which path she takes. Bob enters the cave, flips a coin and shouts the outcome. If Alice knows the magical word she opens the door, if necessary, and returns along the path chosen by Bob. If she lied about knowing it, then she has a 50% chance of returning through the correct path (*i.e.* by guessing Bob's outcome). If they repeat this protocol multiple times, the chance of Alice tricking Bob decreases. Thus, if Alice always exits through the right path, Bob can conclude that Alice really knows the secret word.

[20] at least twice as large as the picture in each dimension

[21] At least the hole should be covered while the book is pulled out.

**“Locked Boxes” Solution.** A classical method for explaining symmetric encryption is through the use of “impenetrable” locked boxes (see [4, 5]). More precisely, Alice and Bob both have a copy of the key that opens a chest. To exchange messages, Alice simply puts her letter in the box, locks it and sends it to Bob. Since Bob has an identical copy of the key, he opens the chest and reads the letter. Another protocol that can be explained using locked boxes is Shamir’s three-pass protocol [14]. First, Alice puts her message in a box, locks it with her private padlock and sends it to Bob. Then, Bob places his private padlock on the box and sends it back to Alice. Once she receives the box, she removes her padlock and sends the box to Bob. Finally, Bob removes his padlock and reads Alice’s message. In order to popularize cryptography to non-specialized audiences, the authors of [4] used a toolbox or a loose chain to implement the previous physical example of Shamir’s protocol. The authors point out it is easy to prove[22] to audiences that a persistent code-breaker could always dismantle a padlock, or X-ray it, and hence crack the code (*i.e.* knowing the inside of the lock is isomorphic to knowing the key). Thus, we have to employ other techniques than the secrecy of the encryption method.

By relaxing the security requirements from an “impenetrable” box to a tamper-evident box (*i.e.* the receiver can detect if someone managed to open the box) the authors of [15, 16] devise a series of secure protocols.

**Ciphers Based on a Deck of Cards.** Schneier designed the “Solitaire” cipher [20] for the book “Cryptonomicon” [23][23]. Solitaire was intended to be the first truly secure “pen and paper” cipher. It requires only a pack of cards both for encryption and decryption. A similar example is the “Mirdek” cipher [7].

**“PEZ Dispenser” Solution.** In [3] the authors present a solution for voting using a PEZ dispenser. Consider a group of kids wishing to vote between two candidates without revealing anything except the final outcome. Assume that they have a PEZ dispenser, which may be previously loaded with some publicly known sequence of red and yellow candies. The kids take turns. Each one decides how many candies to pop out of the dispenser according to his vote. Note that no other kid can see the number or the colors of these candies. Also, it is forbidden for the participants to weight the dispenser and, thus, deduce the number of remaining candies. When this process ends, the color of the candy on top has to correspond to the correct majority vote. The voting process is completed when one of the kids pops an additional candy and announces its color.

**“Phonebook” Solution.** Khovanova recalls on her blog [13] that, for explaining one-way functions, Micali used the following example of encryption. We start by assuming that Alice and Bob obtain the same edition of the white pages book for a particular town. For each letter Alice wants to encrypt, she finds a person in the book whose last name starts with this letter and uses his/her phone number as the encrypted version of that letter. To decrypt the message Bob has to

[22] *e.g.* by showing a sawn up padlock

[23] entitled “Pontifex” in the book

read through the whole book to find all the numbers. The decryption will take a lot more time than the encryption. Unfortunately, the technology changes and the example is not up to date anymore: reverse look-up is always possible in a digital world. Furthermore, regarding the security of the scheme, an 8th grader said: “If I were Bob, I would just call all the phone numbers and ask their last names.” A similar example may be found in [4]. Such examples are very good for teaching one-way functions to non-mathematicians.

**“Colors” Solution.** The Diffie-Hellman protocol can be depicted using colors as further presented. An illustration using common paint may be found in [2]. The idea, first proposed by Simon Singh [22], relies on two properties of colors: ① it is easy to mix two colors and ② given a color that was obtained by mixing two other colors, it is difficult to reverse the process[24]. As a specific example, we may assume that yellow 🟨 is a public color. Let us further consider that Alice’s secret color is blue 🟦 and that Bob’s secret color is red 🟥. The parties wish to agree on a new shared secret color. In the first step, Alice sends green 🟩 to Bob (*i.e.* the result of yellow 🟨 mixed with blue 🟦). Then, Bob sends orange 🟧 to Alice (*i.e.* the result of yellow 🟨 mixed with red 🟥). By mixing the received color with the secret color, each party obtains the common secret brown 🟫 (*i.e.* Alice mixes orange 🟧 with her blue 🟦 and Bob mixes green 🟩 with red 🟥).

Although insecure[25], the digital version of the above protocol is a good teaching tool *e.g.* when trying to explain beginners how to use colors in the case of programming languages used in web development.

## B Physical Public Key Encryption

We further present a generic protocol based on the protocols described in [11]. Alice and Bob have access to a physical medium characterized by a parameter $p(t)$, such that $p(t)$ has two components $p = p_a(t) \circ p_b(t)$, where $\circ$ is a group law and $p_a(t)$, $p_b(t)$ can randomly be changed by varying $t$. In her private spaces $U$ and $V$, Alice and Bob secretly vary $p_a(t)$ and, respectively, $p_b(t)$. Note that Eve only has access to $p(t)$. First Alice and Bob randomly vary $p_a(t)$ and $p_b(t)$. When they agree to synchronize[26], Alice and Bob stabilize their parameters $p_a(t') = a$ and $p_b(t') = b$. Bob can measure $p(t') = a \circ b$ and deduce Alice’s value $a$. Similarly, Alice can compute $b$.

**Example.** We consider the setup from Section 2.3. Thus, the components that Alice and Bob vary are their corresponding speeds values $a$ and $b$. Once the system is stabilized Bob can deduce $a$ using the attack we described in Section 2.3, but Eve can only deduce $b - a$.

[24] and obtain the initial colors

[25] When mixing two colors which can be described in the RGB (Red-Green-Blue) color model one can revert the process due to the uniqueness of each color. Note that such a phenomenon does not happen when working with paint.

[26] through the use of an authenticated channel