# How to Explain Modern Security Concepts to your Children [*]

Xavier Bultel[1], Jannik Dreier[2], Pascal Lafourcade[1,3], and Malika More[1,3]

[1]*LIMOS, Université Clermont Auvergne,* `firstname.lastname@udamail.fr`
[2]*LORIA, Université de Lorraine,* `jannik.dreier@loria.fr`
[3]*IREM de Clermont-Ferrand, Maison pour la Science en Auvergne*

**Abstract**

At the main cryptography conference CRYPTO in 1989, Quisquater *et al.* published a paper showing how to explain the complex notion of zero-knowledge proof in a simpler way that children can understand. In the same line of work, we present simple and intuitive explanations of various modern security concepts and technologies, including symmetric encryption, public key encryption, homomorphic encryption, intruder models (CPA, CCA1, CCA2) and security properties (OW, IND, NM). The explanations given in this paper may also serve in demystifying such complex security notions for non-expert adults.

---

# 1 Motivation and Contributions

All the technical concepts and notions explained in this paper are well-known to cryptographers, and occur in most introductory courses for modern cryptography. The goal of this paper is to propose explanations and illustrations that can be used to explain these main modern security notions in a simple and understandable manner to non-experts of all sorts: children, students, adults without background in mathematics or computer science, and so on.

The paper is the result of more than ten years of teaching security courses at different levels from primary school to master courses at university. Over time, a set of helpful illustrations and examples for the various concepts have emerged, as well as a certain structure in the presentation. The paper follows this structure: We start with our illustrations for symmetric encryption and asymmetric or public key encryption (PKE). Then we give our illustrations of the different security levels that cryptographic systems can achieve, starting with the different intruders (CPA, CCA1 and CCA2), and followed by the different security properties One-Wayness (OW), Indistinguishability (IND), Non-Malleability (NM) and Key Anonymity (KA). We finish by giving intuitions for Homomorphic Encryption (HE), cryptographic hash functions, signatures and the Diffie-Hellman key exchange.

**Related Work:**   There is an international effort on the dissemination of computer science[1], in particular towards children. For instance, the authors of the Computer Science Unplugged group have elaborated several activities presenting various concepts of computer science. Among them, they provide several cryptographic activities like for instance the *Peruvian Coin Flip*, originally proposed in [Fellows and Koblitz, 1993], where people who do not trust each other and cannot see each other are able to agree on the outcome of a random coin flip. They also propose a *Kid Krypto* public key encryption scheme that allows children to encrypt and decrypt messages without having to share a secret key, using a graph. Another activity, called *Sharing Secrets* and also introduced in [Fellows and Koblitz, 1993], shows how to share personal information accurately without having to give up any privacy at all. In addition, the "*Scout Patrol*" activity presents a symmetric key encryption originally presented in [Fellows and Koblitz, 1993]. In the "CryptoClub"[2] after-school program, middle-grade students explore cryptography while applying mathematics to make and break secret codes. This program is supported by the University of Illinois at Chicago. Since 2010, the *National Security Agency* (NSA) also has its own pro-

---

[1] `http://csunplugged.org`
[2] `http://www.cryptoclub.org/`

gram for kids[3]. There are several other websites[4] that offer secret messages to challenge kids. Purdue University[5] proposes a web page that collects several resources for explaining cryptology to kids. Currently, the cryptographer Jacques Stern oversees a French nation-wide cryptanalysis competition for kids called *Alkindi*[6], which is the name of a famous Arabic mathematician that did the oldest known cryptanalysis.

Our work follows lines of Quisquater *et al.* [Quisquater et al., 1989] where the authors explain the concept of zero knowledge proofs. They use "*the strange cave of Ali Baba*" as an illustration. This circular cave has a door protected by a secret code. They show how, without revealing her code, Peggy the prover, can convince Victor, the verifier, that she knows the code. In [Beaver, 2009], the author proposes to introduce zero knowledge proofs in a classroom of undergraduate students setting using the cave example and also to study the Fiat-Shamir protocol [Fiat and Shamir, 1987]. However the activity he proposes requires some advanced mathematical notions. In [Gradwohl et al., 2007] a simple sudoku zero knowledge proof using card games is presented. In [Bultel et al., 2016], using cards and envelopes, four physical interactive zero knowledge proofs are proposed for four Japanese logic games: Akari, Kakuro, KenKen and Takuzu.

In [Bell et al., 2003] the authors propose to explain digital signatures, digital cash, on-line poker and secure voting to school children. In [Borovik, 2002], the author gives an implementation of activities proposed in Kid Krypto. More recently in his blog [Green, 2014], M. Green proposes an understandable explaination of the zero-knowledge concept. In [Moran and Naor, 2010], the authors use sealed envelopes, locked boxes and scratch cards to construct secure oblivious transfer, bit-commitment and coin flipping. In [Fellows and Koblitz, 1994] and [Koblitz, 1997], the authors use several combinatorial objects to explain cryptographic notions to children (and non-expert adults). Using high school mathematics and more complex mathematical notions, they present one-way functions, hash functions, signatures, public key cryptography (thanks to the "*Peruvian coin flip*" activity and to a cryptosystem based on a graph similar to Kid Krypto). In [Naor et al., 1999], Naor *et al.* explain to kids how to convince people that you know "where is Waldo" without revealing any information about his position. These activities use simple material that kids can manipulate and help them to understand the challenges behind the construction of a secure primitive. In [Caballero-Gil and Bruno-Castaneda, 2007], the authors propose for secondary school activities to teach mathematics through cryptography. They present the

---

[3]https://www.nsa.gov/kids

[4]http://www.thunk.com

[5]http://www.cerias.purdue.edu/education/k-12/teaching_resources/lessons_presentations/cryptology.html

[6]http://www.concours-alkindi.fr/

*Vigenère* cipher to introduce the notion of functions, a knapsack cipher to introduce vectors, and to present randomness, proofs and equations. They use oblivious transfer, zero knowledge protocols, and the Shamir secret sharing threshold scheme. In [Phan, 2005], an attempt to explain block cipher cryptanalysis to kids is presented. In [Forišek and Steinová, 2012], some metaphors and analogies for teaching some computer science algorithms are proposed. As it has been shown by A. Finkel in [Arnoux and Finkel, 2010], proposing mental representations helps students to learn abstract notions. One of our aims is to propose mental representations to help people understand modern security notions. Another goal is to enable children to understand how security works and to show them that mathematics is used to realize "fancy" modern technology. This paper also shows that security is not easy to obtain.

**Outline:**   In Section 2, we present our illustrations for the two main notions of encryption: symmetric and asymmetric encryption. Then, in Section 3, we explain our analogies for the security notions used for encryption schemes. We give our illustrations for homomorphic encryption in Section 4, followed by cryptographic hash functions in Section 5, signatures in Section 6 and the Diffie-Hellman key exchange in Section 7. In Section 8, we conclude our paper and present some perspectives.

# 2   Illustrating symmetric and asymmetric encryption

All encryption schemes invented before the Second World War are *symmetric* encryptions, which means that a secret key is shared between two persons, and this key is required to both encrypt and decrypt a message. Using advanced mathematics, R. Rivest, A. Shamir and L. Adelman were able to propose the first *asymmetric* or *public key encryption scheme* [Rivest et al., 1978b]. In this case, the encryption and decryption stages are not symmetric, and do not use the same key. The encryption key is public, and can be used by anybody, whereas the decryption key is secret, known only to the recipient of the encrypted messages. This conceptual revolution allows anybody to use the public key of Alice to send encrypted data to her, that only she is able to decrypt using her private key.

## 2.1   Symmetric Key Encryption Schemes

To illustrate symmetric key encryption, we use the following analogy based on a lockable chest and keys as shown in Figure 1.

**Illustration 1** (Symmetric Encryption)**.** *If Alice and Bob have the same key that opens the chest, then they can exchange secret letters. Alice puts her secret letter,*

*called a* plaintext*, in the chest and closes it with her key. This is called a* cipher. *Then she sends it to Bob. Since Bob has the same key as Alice, he can open the chest and get the letter. This is the principle of symmetric encryption: a symmetric encryption uses the same key to encrypt and to decrypt a message. This key is called a* symmetric key.



Figure 1: Symmetric encryption.

To illustrate how padlocks and keys can be constructed in the digital world of computers, a simple example we often use is the Caesar Cipher, where each letter is shifted three letters to the right in the alphabet. In this case the symmetric key used is $3$, and the set of possible keys is $\{1, \ldots, 25\}$.

The Caesar cipher can also be used to illustrate weaknesses and attacks, as a build-up towards the security definitions explained below: often students see directly that such encryption scheme can be broken by testing all possible shifts, which consists in trying only $25$ possibilities. This introduces the notion of *brute-force attacks*, and illustrates the necessity for a large key space. Since here there are only $25$ possibilities, the exhaustive search always succeeds.

Sometimes they also notice that, in English or any other language, some letters are more frequently used than others. Then, by counting the most frequent letters in the ciphertext, provided that it is long enough, it is possible to reduce the number of possibilities to try (the so-called *frequency analysis*).

To show that not all encryption schemes are vulnerable to attacks such as frequency analysis, a useful example is the One Time Pad Encryption (OTP) (also known as *Vernam encryption* because it is generally credited to Gilbert S. Vernam and Joseph O. Mauborgne in 1917, but it has been shown in [Bellovin, 2011] that it was actually invented 35 years earlier by Frank Miller). This encryption scheme is perfectly secure, *i.e.*, according to Shanon's information theory, without knowing the key no information about the message is leaked [Vaudenay, 2005, Baigneres et al., 2010]. In this scheme, the encryption of the binary message $m$ with the binary key $k$ is $m \oplus k$, where $\oplus$ is the exclusive-or operator (in other words a bit-wise addition modulo 2), which means that the size of the key $k$ is the same as the size of the message $m$, which requires to have a large shared secret. For instance, if the message $m = 010111$ is encrypted with the key $k = 10010$,

then the ciphertext is $c = 100101 = m \oplus k$. The decryption consists in computing $c \oplus k = m$ using the same key $k$. To allow kids to encrypt messages using this scheme, it is possible to use the OTP principle with normal letters from the alphabet and an addition modulo 26 instead of a binary representation.

Moreover, key re-usage can have dramatic consequences on the security of the communication, as the following 3-pass Shamir protocol illustrates. It aims at exchanging a secret message $m$ without sharing a key beforehand. Formally the protocol is composed of three messages, where $\{m\}_{k_A}$ denotes the encryption of $m$ with the secret key $k_A$ of Alice:

1. $A \rightarrow B : \{m\}_{k_A}$

2. $B \rightarrow A : \{\{m\}_{k_A}\}_{k_B}$

3. $A \rightarrow B : \{m\}_{k_B}$

Note that this protocol works only if the encryption has the following algebraic property, called *key commutativity*: $\{\{m\}_{k_A}\}_{k_B} = \{\{m\}_{k_B}\}_{k_A}$.
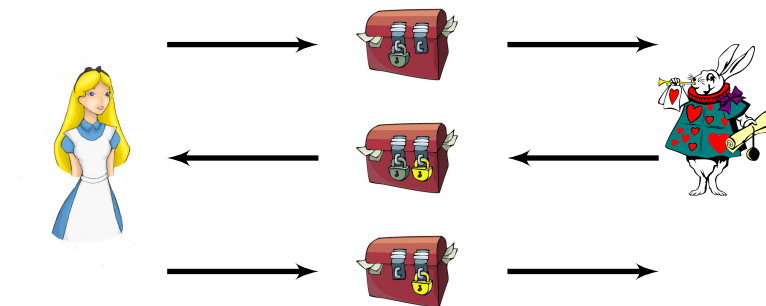


Figure 2: 3-pass Shamir protocol.

**Illustration 2.** *With kids, this protocol can be realized using a real box (with two places for padlocks), two padlocks, and regular mail, as described in Figure 2 and first proposed in [Shamir et al., 1981]. The kids have to find a secure way for Alice to exchange a secret message with Bob, without sharing a key. The solution consists in the following protocol: Alice puts her padlock on a box containing the secret message and sends it to Bob. Then Bob adds his own padlock on the box and sends it back to Alice. Now Alice removes her padlock and sends to Bob the box that only has the Bob padlock. Finally Bob removes his padlock and opens the box to discover the secret message.*

This simple protocol can be used to illustrate protocol attacks, as there is for example a *reflection attack*. Suppose an intruder intercepts the first message and

simply sends it back to Alice. Then Alice removes her key and sends the plaintext message $m$ to the intruder. This is due to the lack of *authentication* in this protocol. It is also possible to mount a *man-in-the middle attack*, where that intruder first plays a full session with Alice and learns the message $m$. Then he does a second full session with Bob to let him know the message $m$. In that case Bob could believe that only he and Alice know $m$, but the intruder also knows it.

Using OTP with this protocol can be used to create an interesting exercise. For this, we need to consider the four algebraic properties of the exclusive-or operator: $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity), $x \oplus y = y \oplus z$ (Commutativity), $x \oplus 0 = x$ (Unit element), and $x \oplus x = 0$ (Nilpotency). The goal of the exercise is to ask the students to discover the message $m$ just by listening to the three messages of the protocol. Note that the One Time Pad (OTP) encryption is commutative since: $\{\{m\}_{k_A}\}_{k_B} = (m \oplus k_A) \oplus k_B = (m \oplus k_B) \oplus k_A = \{\{m\}_{k_B}\}_{k_A}$ and could thus be used here. To help them, it is possible to recall that the intruder collects the following three messages during the execution of the protocol: $m \oplus k_A$; $(m \oplus k_A) \oplus k_B$; $m \oplus k_B$. Then they might discover that the intruder can learn $m$ just by performing the exclusive-or of these three messages, since $m = m \oplus k_A \oplus (m \oplus k_A) \oplus k_B \oplus m \oplus k_B$. The box analogy reflects the fact that the encryption is secure and key commutative, however this analogy does not take into account some algebraic properties of the encryption scheme that might lead to some even simpler attacks.

## 2.2 Public Key Encryption Schemes

We first explain the notion of *Public Key Encryption* (PKE), then we discuss how to present the RSA and ElGamal encryption schemes as examples.

**Illustration 3** (Public-Key Encryption). *The principle of a public key encryption scheme is that everyone can get the encryption key of somebody. These keys are often published on web page of their owner and/or stored in some public key storage like for instance the MIT PGP Public Key Server[7]. These encryption keys can be seen as open padlocks that are distributed without the associated keys, as shown in Figure 3, and proposed in [Beutelspacher, 1994]. Using these padlocks, called* public keys, *everyone can encrypt a message but only the owner of the corresponding decryption key (so-called* secret key) *can open the padlock and recover the plaintext. The public key encryption schemes are also called* asymmetric encryption schemes *by opposition to the symmetric encryption schemes since they use a public key for encryption and a secret key for decryption.*

The notion of *trapdoor one-way function* is the main concept of public key cryptosytems. A *one-way function* is a function that is easy to compute but which
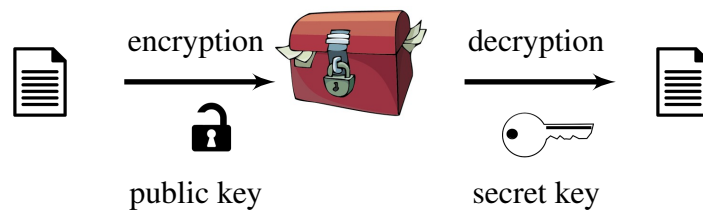
---

[7]https://pgp.mit.edu/

Figure 3: Asymmetric encryption scheme

inverse is computationally hard to compute. To illustrate the notion of one-way function one can use a phone book. In this situation, it is easy to find the phone number knowing the name of a person, but it is quite difficult to recover the name of the owner of a phone number knowing only this information[8] . A *trapdoor one-way function* is a one-way function such that knowing some additional information, the *trapdoor*, the inverse becomes easy to compute. The trapdoor for the phone book example could be a reverse phone directory, which is easy to compile for the person who created the original directory. In public key cryptography the trapdoor corresponds to the secret key which is difficult to find, given only the public information. This secret key allows his owner to decrypt encrypted messages computed using the public encryption function.

The concept of public-key encryption schemes can be illustrated both using the RSA and ElGamal schemes.

**Kid Krypto Encryption Scheme.**   Most real life public key encryption systems use somehow sophisiticated mathematics that are not suitable for kids before high school. The Kid Krypto activity uses only elementary school mathematics and was proposed in [Fellows and Koblitz, 1993] and used by the Computer Science Unplugged group[9]. Despite its security flaw, it can be used in the classroom.

It relies on the difficult (*i.e.* NP-complete) graph problem called *Minimum Dominating Set*. A dominating set of a simple graph is a subset of vertices such that any vertex of the graph is either in the subset or adjacent to a vertex in the subset, as it shown in Figure 4. Since the set of all vertices is obviously a dominating set, the question is to find a dominating set containing as few vertices as possible.

As a first step, students have to be convinced that it is actually difficult to find a minimum dominating set on a graph by working out several examples. The

---

[8]The phone book example is generally attributed to Silvio Micali [Khovanova, 2010] and Arto Salomaa [Bauer, 1997].
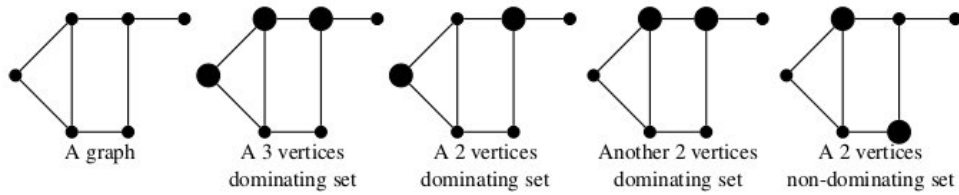
[9]http://csunplugged.org
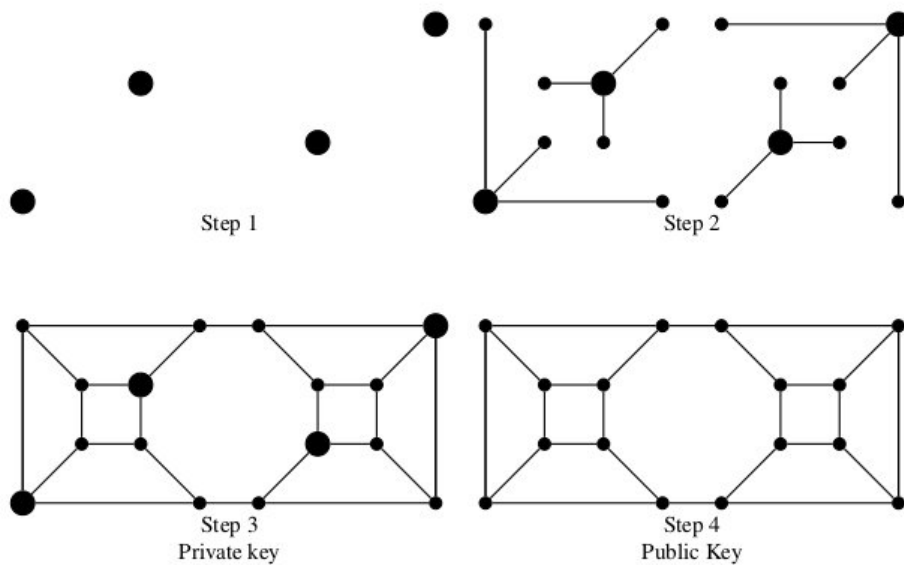
Figure 4: Dominating Set



Figure 5: Keys

authors present a method to build such puzzles, as it shown in Figure 5: draw some vertices. This set of vertices is the minimum dominating set. To each of them, add some dominated vertices, forming disconnected star graphs. Finally, hide your connected components by adding some edges between dominated vertices.

Let us now explain how to encrypt and decrypt a number using this graph as a public key, as it shown in Figure 6. Note that this does not work for every graph with a minimum dominating set, but only for "*Perfect Code*" graph as defined by Fellows and Koblitz in [Fellows and Koblitz, 1993]. The private key is the minimum dominating set. Encrypting a number is done in two steps. First, the number is randomly decomposed into a sum of numbers and spread over the vertices of the graph, so that the values of all vertices add up to the initial value. Then, the
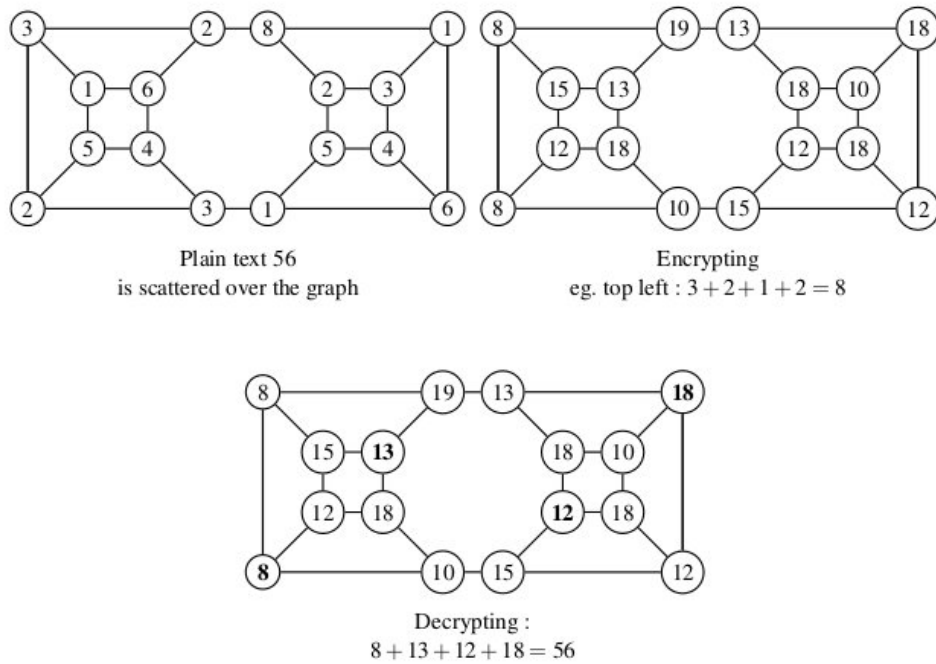
9

Figure 6: Encryption and Decryption

value of every vertex is replaced by the sum of its original value plus that of all its neighbours. The cipher is the resulting graph, with the obtained values on the vertices. Decrypting the message consists in adding the values of the dominating vertices forming the secret key.

If encrypting and decrypting might be somehow tedious, however this activity requires only few mathematical skills. Finally, note that this scheme is not actually secure, because the encrypting step can be described using linear equations, that are easily (*i.e.* polynomially) solved, using for instance Gaussian elimination.

**RSA Encryption Scheme.**   In 1978, Ron Rivest, Adi Shamir and Leonard Adleman [Rivest et al., 1978b] designed the first public key encryption scheme called RSA for Rivest, Shamir, Adelman. Their encryption scheme relies on the computationally hard problem of *factorization*: given $n$, the product of two primes numbers $p$ and $q$, we do not know up to now any efficient algorithm to recover $p$ and $q$, whereas it is easy to compute $n$ from $p$ and $q$. The public key is composed of $e$ and $n$, where $e$ is an integer that is coprime to $(p-1) \cdot (q-1)$ and $n$ is the product of the two primes $p$ and $q$. The secret key is $d$, the inverse of $e$ modulo $(p-1) \cdot (q-1)$. The encryption of the message $m$ is $m^e \mod n$ and

10

the decryption of the ciphertext $c$ is $c^d \mod n$. The correctness of this encryption scheme relies on the little Fermat's theorem. Although the underlying theory is complex, children can easily execute the encryption and decryption using small numbers and pocket calculators or even online resources like Sage[10] or Wolfram Alpha[11], as there are only simple operations such as multiplication, exponentiation and modulo. Moreover one can convince them that factorization is difficult for large numbers by giving an example of a large number and asking them to find the factors.

Moreover, RSA can be used to illustrate the difference between deterministic and randomized encryptions. It is easy to see that RSA is *deterministic*, which means that the encryption of a message $m$ with a given public key always produces the same ciphertext. To illustrate that this can be a problem, one can sketch a very simple voting protocol where voters encrypt either "yes" or "no" using the public key of a voting authority: if the encryption is deterministic, it is easy to find out how each voter voted by simply encrypting "yes" and "no" and comparing the result to the voter's submissions.

**ElGamal Encryption Scheme.**     This encryption scheme was invented by T. ElGamal in 1985 [ElGamal, 1985]. It is a randomized encryption, which means that each encryption of a given message $m$, using a given public key, produces a different ciphertext. This encryption scheme relies on the hard problem called *discrete logarithm*: knowing $g$, $g^a \mod p$ and $p$ it is difficult to find $a$ but it is easy from $g$, $a$ and $p$ to compute $g^a \mod p$. The public key is composed of $g$, $p$ and $h$ such that $p$ is prime, and $h = g^a \mod p$. The secret key is $a$. The encryption of a message $m$ is the couple $(g^r \mod p, h^r \cdot m \mod p)$, where $r$ is a random number. The decryption of a ciphertext $c = (c_1, c_2)$ consists in computing $c_2/c_1^a \mod p$.

ElGamal has the advantage of relying on relatively simple math, so that for example high school students can verify the correctness of the decryption.

Moreover, it is easy to see for non-experts that in RSA and Elgamal encryption schemes, the security level depends on the length of the key: one can for example ask them to factorize a small number, and use this to break RSA by decrypting a message without using the secret key. Note that there are several other public key encryption schemes that have been proposed, based on different techniques and different computationally hard problems [Schneier, 1996].

---

[10]http://www.sagemath.org
[11]https://www.wolframalpha.com/

# 3 Introducing Encryption Secheme Security Levels

At this point, the children have some intuition on how different schemes work and know some classical attacks. Moreover they have seen that some schemes might be secure in some contexts but insecure in others, and that the choice of parameters is important. This allows us to motivate the necessity for precise security definitions to evaluate the security level of encryption schemes. For this, one has to define the intruders abilities and the security properties. We start by presenting the various intruders and then the different properties. Many relationships between these notions have been established in [Bellare et al., 1998]. Moreover similar results exist for symmetric encryption schemes [Bellare et al., 1997].

## 3.1 Illustrating Intruders

We now present the main intruders against which encryption schemes should try to resist. We start with a basic intruder and then increase its capabilities up to a powerfull intruder.

### 3.1.1 Basic Attacker (BA)

**Illustration 4** (Basic Attacker (BA)). *Such attacker that tries to recover the plaintext given only a ciphertext (produced either by a symmetric or assymmetric encryption scheme), called* challenge*. This is illustrated in Figure 7: on the right we have the intruder, and the* challenge *is pictured as the locked box on the left.*
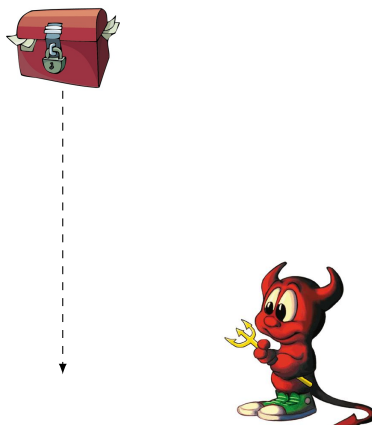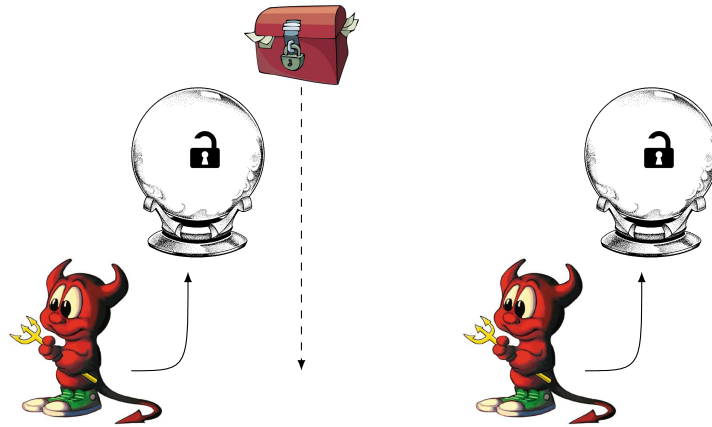


Figure 7: Basic adversary.

Figure 8: CPA adversary.

Having an encryption scheme that resists to such an intruder is the minimum security expected from an encryption scheme. In 1984, M. Miccali and S. Goldwasser proposed a more realistic intruder and considered a more powerful security model. In 2014, they received the Turing Prize for their work about the notion of *Chosen Plaintext Attacks* defined in [Goldwasser and Micali, 1984]. Later on, even more powerful intruders were introduced [Naor and Yung, 1990, Rackoff and Simon, 1993]. Our aim is to present these intruders in an intuitive way. To illustrate the intruders the examples given in Section 3.2 can be used.

### 3.1.2 Chosen Plaintext Attack (CPA)

The idea is very simple: when studying the security of an encryption, knowing several pairs of plaintexts and ciphertexts can help to break the encryption. A famous example is how A. Turing and his team broke the ENIGMA machine during the second World War.

**Illustration 5** (Chosen Plaintext Attack (CPA)). *This notion is illustrated in Figure 8 as follows. On the left, we have the intruder before receiving the challenge, and on the right the intruder after receiving the challenge (as above, the challenged is pictured as the locked box). In the case of chosen plaintext attacks, before and after receiving the challenge, he can encrypt some messages, pictured as an open padlock inside a crystal ball. In cryptography, these "crystal balls" that allow the adversary to access certain values or to perform certain operations are called* oracles.

Note that such an intruder is more powerful than an intruder that only knows the challenge, *i.e.,* only one ciphertext: in the example of the voting system using
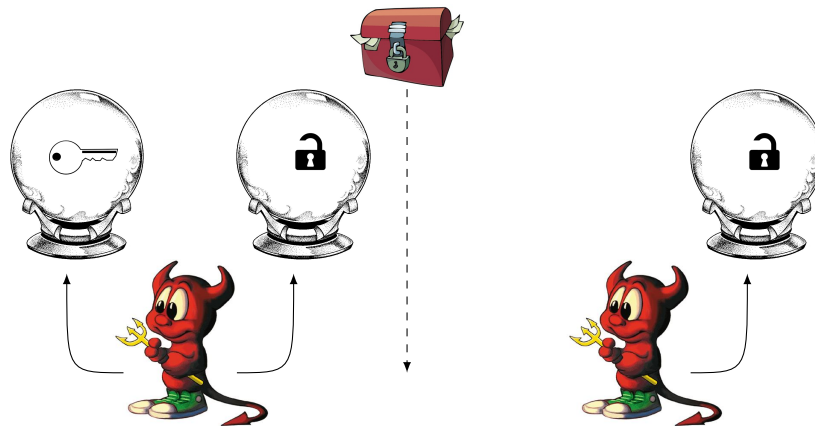
13

Figure 9: CCA1 adversary.

RSA encryption given above the attacker could "break" the secrecy of the votes by simply encrypting all possible votes. Thus having a cryptosystem that resists to a CPA intruder is more secure than only resisting to the basic attacker that only knows the challenge, and cannot encrypt values of his choice.

### 3.1.3 Non-Adaptive Chosen Ciphertext Attack (CCA1)

**Illustration 6** (Non-Adaptive Chosen Ciphertext Attack (CCA1))**.** *In 1990, M. Naor and M. Yung [Naor and Yung, 1990] introduced an even more powerful attack model, also known as the "Lunchtime Attack". They consider a scenario where an employee has secret keys on his computer, which he uses to receive encrypted messages. Anyone having access to his computer is able to decrypt all previously received messages, assuming that the computer stores the secret key. Now suppose that the employee goes for lunch but forgets to lock his computer: an attacker can access it during a limited period of time. In particular, he can open all previously received encrypted messages, and decrypt other messages of his choice. After lunchtime, the adversary locks the employee's computer in order not to raise any suspicion.*

*Then the adversary tries to decrypt the next message received by the employee. In other words, the new adversary model consists in an adversary who can encrypt messages and additionally has access to a decryption oracle, i.e., a functionality that allows him to decrypt a ciphertext of his choice,* **multiple times before he gets the challenge***. This intruder is known as CCA1 for* Non-Adaptive Chosen Ciphertext Attack*, and illustrated in Figure 9: the key inside the leftmost crystal ball represents the possibility to decrypt, to which he only has access before receiving the challenge.*

14

### 3.1.4 Adaptive Chosen Ciphertext Attack (CCA2)

**Illustration 7** (Adaptive Chosen Ciphertext Attack (CCA2)). *In 1993, C. Rackoff and D. Simon [Rackoff and Simon, 1993] imagined an even more powerful adversary. This adversary can encrypt messages, and has access to a* **decryption oracle multiple times before and AFTER he gets the challenge**, *but of course he cannot decrypt the challenge. This adversary is known as the* Adaptive Chosen Ciphertext Attack, *or "CCA2" for short, and illustrated in Figure 10.*
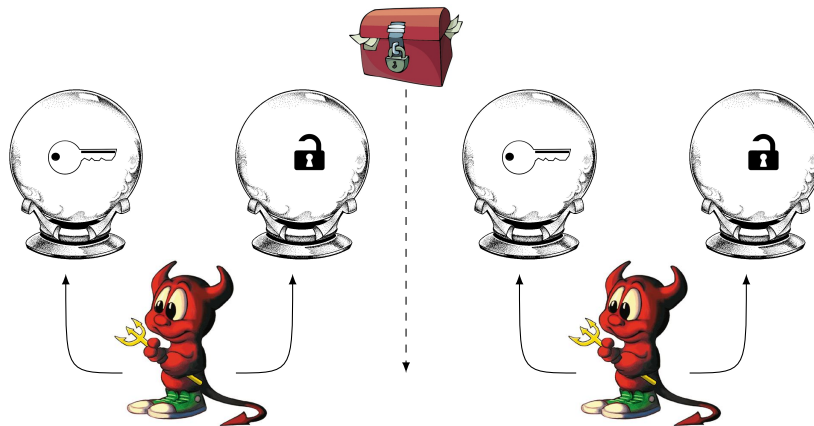


Figure 10: CCA2 adversary.

Formally modeling all these notions requires the use of Probabilistic Polynomial Time Turing machines (PPTs) and security games that exceed the abilities of our young target audience. However, as we showed it is possible to present the different types of intruders and to describe the differences between them.

## 3.2 Security Notions

We now present some security notions that have been introduced simultaneously with the adversary models presented before, where each security notion corresponds to one of the adversaries explained above. For this purpose, we developed a metaphor using different types of containers to model the security properties of cryptographic primitives. In [Bellare et al., 1998], Bellare et al. formally established the relationships between these security notions and the above intruder types.

### 3.2.1 One-Wayness

**Illustration 8** (One-Wayness (OW)).

15

Figure 11: Representation of the One-Way property.

*The first security notion is called* One-Wayness *(OW). For an adversary, given a ciphertext, it should be difficult to recover the plaintext in a reasonable amount of time without knowing the associated decryption key. We represent this kind of encryption as a translucent plastic bag (it is not possible to read the message) that is closed with a padlock (representing the one-way encryption) and contains the secret message.*

A one-way encryption scheme is secure against the basic attacker as his only possibility is to directly try to decrypt the message, but it is not secure against stronger adversaries as explained below.

### 3.2.2 Indistinguishability

An encryption scheme is said to have the *Indistinguishability* (IND) property if an intruder that knows two different plaintext messages having the same size and a challenge, which is the encryption of one of the two messages, cannot determine which one is encrypted in the challenge. In [Goldwasser and Micali, 1984], the authors introduced this notion.

**Illustration 9** (Indistinguishability (IND))**.**



Figure 12: IND-CPA attack on OW encryption.

*We start by showing that a one-way encryption primitive can leak some information in some situations (see Figure 12). We have two messages: one is written on a gray paper and the second one is written on a white paper. These two messages plus one challenge (the translucent plastic bag) that contains one of the two*

16

*previous messages, are given to the adversary. Is it possible for the adversary to determine which message has been encrypted in the challenge?*[12] *Usually kids have the intuition that since the bag is translucent it might be possible to determine the color of the message inside, if not to read the plaintext itself. Asking the kids how to prevent such an attack, they naturally propose to put the message in a black bag (see Figure 13). Continuing our metaphor, this image represents the notion of indistinguishability.*



Figure 13: Representation of indistinguishability property.

Some students are able to describe the following indistinguishability attack with a CPA intruder on a one-way public-key encryption scheme: encrypt with the public key one of the two messages and compare the result with the challenge, if they are equal you know which one has been encrypted, otherwise it is the other message that is inside the challenge. This attack is general and works for all deterministic encryption schemes, like for instance RSA as noted above.

In Figure 14, we show the encryption of a padlock image using the symmetric encryption AES, first using Electonic Code Book encryption mode, and in the second image using Cipher Block Chaining encryption mode. The ECB mode is only one-way while the CBC mode is IND-CPA. This example is often convincing that the translucent bag is a good representation of the notion of OW.

### 3.2.3   Non-Malleability

An encryption scheme is said to have the *Non-Malleability* (NM) property if an intruder that knows a ciphertext (the challenge) cannot construct other valid ciphertexts for which he knows a relation between the corresponding plaintexts and the plaintext encrypted in the challenge.

**Illustration 10** (Non-Malleability (NM))**.**

---

[12]As we have seen for the vote example with CPA adversary, an encryption that is used in a vote system should have the indistinguishability property in order to avoid that an adversary determines the vote of each voter.
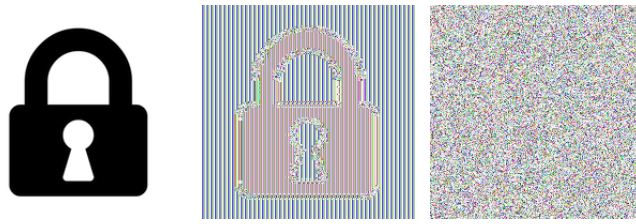
Figure 14: Encryption with a OW-CPA encryption and an IND-CPA secure encryption.



Figure 15: Attack against IND-CCA2.

*We illustrate his property by describing an attack on a scheme that does not have the NM property. If we consider a CCA2 intruder, then the black plastic bag is not safe. In order to help students to discover the corresponding attack, we propose a situation where the message is written on a Rubik's Cube and placed in a black plastic bag.*

*If the students remember correctly the capabilities of a CCA2 intruder, they are able to propose the following attack: turn the Rubik's Cube to modify the challenge and give the new ciphertext to the decryption oracle. The oracle decrypts the modified message and gives it back to the intruder i.e., the oracle opens the bag. He just has to turn back the cube to recover the plaintext corresponding to the challenge. In order to prevent such attack, a more secure encryption can be represented as a rigid back box.*

*Such a rigid black box illustrates the notion of* non-malleability. *In other words it is not possible for the intruder to modify the challenge to obtain a valid ciphertext that has a pre-established relationship with the plaintext encrypted in the challenge. This notion has been formally introduced by Dolev et al. in 2000 [Dolev et al., 2000].*

Figure 16: Representation of Non-Malleability property.

### 3.2.4 Key Privacy

The previous security properties focus on the security of the messages, but in 2001, cryptographers also introduced a property concerning the security of the key used in a cryptosystem [Bellare et al., 2001]. The idea is similar to the indistinguishability of the message, only that this time it concerns the key. The adversary selects a message $m$, and a pair of public keys. The challenge is composed of the message $m$ encrypted with one of the two keys. The goal for the intruder is to determine which key has been used.

**Illustration 11** (Key Privacy)**.**



Figure 17: Attack on Key Privacy property.

*Using an ordinary padlock and looking at the shape of the key hole, one can probably determine which key has been used. Note that the original message is part of the challenge. Such an encryption scheme does not preserve the* key privacy. *Figure 18 presents a digital padlock for which it is not possible to distinguish which digital key has been used to encrypt the challenge: such an encryption has the* key privacy *property.*

## 4    Explaining Homomorphic Encryption

In 2009, G. Gentry [Gentry, 2009] solved a long standing open problem introduced by Rivest et al. [Rivest et al., 1978a] that asks for the capability to perform any operation on encrypted data without decrypting the data. Encryption

Figure 18: Key Privacy property.

schemes with this property are extremely useful in the cloud applications. Before 2009, several partial solutions called *partial homomorphic encryption* schemes have been proposed, like for instance [ElGamal, 1985, Naccache and Stern, 1998, Paillier, 1999, Fousse et al., 2011, Goldwasser and Micali, 1984].

**Illustration 12** (Homomorphic Encryption). *G. Gentry presented the problem of the homomorphic encryption in [Gentry, 2010] as follows: Alice has a raw diamond and some gold. She wants a jeweler to make for her a nice ring, but she is afraid that her precious stone is stolen by the jeweler. For this reason, she needs to put the stone in a safe place. Since she also wants the jeweler to work on the stone, she should ask him to use a secure glove box as presented in Figure 19.*
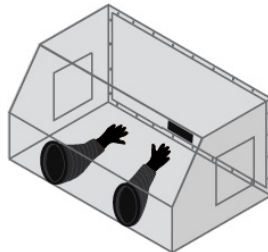


Figure 19: Glove box.

*In other words, the jeweler should be able to apply any function $f$ on the ciphertext (i.e., from outside the safe) and the effect of $f$ appears on the plaintext (inside the safe). More formally, the fully homomorphic property is the following, where $m_1, \ldots m_p$ are messages, $k$ the key and $f$ the function applied outside the encrypted messages and $g$ the function applied inside the encrypted messages:*

$$f(\{m_1\}_k, \ldots, \{m_p\}_k) = \{g(m_1, \ldots, m_p)\}_k$$

Partial homomorphic encryption schemes only allow to perform one operation, like for instance Elgamal [ElGamal, 1985] or RSA [Rivest et al., 1978b], and are a

good first example. For instance for RSA, the product of two encrypted messages $c_1 = m_1^e \mod n$ and $c_2 = m_2^e \mod n$ is equal to $(m_1 \cdot m_2)^e \mod n$ which is the encryption of the product of the two plaintexts $m_1$ and $m_2$.

Moreover, there is a fully homomorphic encryption scheme which does not use sophisticated mathematics: the DGHV encryption scheme [van Dijk et al., 2010]. It only uses modular multiplication and addition and works as follows. The secret key is $p$, an odd number in $[2^{\eta-1}, 2^\eta[$, where $\eta$ is the so-called *security parameter*, which evaluates the security of the scheme. The plaintext is a bit $m \in \{0, 1\}$ and the encryption of $m$ is given by:

$$c = q \cdot p + 2 \cdot r + m$$

where $q$ is a large random number ($q \approx \eta^3$) and $r$ a small random number ($r \approx 2^{\sqrt{\eta}}$), such that $2 \cdot r \geq p/2$.

For decrypting the ciphertext $c$, knowing the secret key $p$, it suffices to perform the two following modulo operations on $c$:

$$m = (c \mod p) \mod 2$$

This encryption scheme is *somewhat*[13] *fully* homomorphic for addition and multiplication (verifying this is a feasible exercise for high school students), hence for all boolean function $f$. The advantage of this symmetric encryption scheme is that it is simple enough to allow us to explain how to store encrypted data in the cloud. Note that it is possible to modify this scheme to encrypt messages longer than one bit.

# 5   Explaining Cryptographic Hash Functions

Cryptographic hash functions are mathematical functions that take as input any bitstring and output a digest of a fixed size, $H : \{0, 1\}^* \to \{0, 1\}^n$. Their desired properties are pre-image resistance, second-pre-image resistance and collision resistance [Rogaway and Shrimpton, 2004], which are explained below. These cryptographic primitives tend to be complex to design. Famous examples include MD5, SHA-1, SHA-256 or SHA-3.

**Illustration 13** (Cryptographic Hash Function). *A cryptographic hash function can been explain using the analogy of fingerprints for humans as shown in Figure 20. Kids can play with ink and fingers, and discover that any object (human) can be reduced to a constant size representation, since each human can be characterized by his fingerprints.*

---

[13]Technically it is not *fully* homomorphic as errors build up over time unless we introduce a bootstrapping.

Figure 20: Representation of an hash function, as a deterministic procedure that takes a human as input and produces his fingerprint as a digest.

*Following this representation of hash functions, the desired properties for cryptography can be illustrated as in Figure 21.*

**Pre-image Resistance:**  *For the fingerprint, given a human, it is easy to obtain his fingerprint, but from a fingerprint it is difficult to recover the corresponding human. This property is close to the one-wayness property for encryption function. More formally given a hash value $h$ it should be difficult to compute a message $m$ such that $h = hash(m)$.*

*In real life the police have a collection of identities and corresponding fingerprints of most of the criminals. By comparing one fingerprint with this collection it is then possible to identify a suspect. To the same goal, cryptographers have invented the so called "rainbow table" that corresponds to an efficient data stucture that contains the list of precomputed hash values of some messages.*

**Second Pre-image Resistance:**  *The idea of this property is that given a human it should be difficult to find another one that has the same fingerprint. More formally, given a input message $m$ it should be difficult to find a different input $m'$ such that $hash(m) = hash(m')$.*

*For instance, the hash function that gives the color of the eyes of a person is pre-image resistant, but not second pre-image resistant, because it is not difficult to find another person with the same color of the eye <span style="color:red">in the population, because there is a small number of eye colors</span>.*

**Collision Resistance:**  <span style="color:red">*A collision pair is two different humans having the same fingerprint.*</span> *If it is difficult to find two different messages that have the same hash value, then the hash function is called* collision resistant*. Fingerprints have this property<span style="color:red">: even true twins do not have the same fingerprint, although they share the same DNA.</span>*

Today hash functions are for example used to check the integrity of a received message. For instance when a software $S$ is downloaded from a server, and $d$ is the digest of this file, *i.e.*, $d = H(S)$, the user can check the integrity of the received file $S'$ by checking that $d$ is equal to $H(S')$. If somebody wanted to

- Pre-image



- Second Pre-image



- Collision



Figure 21: Properties of hash functions.

modify the file by for example including a virus, he would need to find a second pre-image for the given hash value. Moreover, in cryptography, hash functions are also used to reduce the size of some data or to generate random values such as in the Fiat-Shamir heuristic [Fiat and Shamir, 1987].

# 6 Illustrating (Blind) Signatures

Another important primitive in security is the ability to sign documents.

**Illustration 14** (Signature). *A signature attests that you are the author of the document, i.e., the person who has generated it. In real life, this has been used for ages: for instance kings, emperors and presidents all have unfalsifiable seals, as shown in Figure 22. Beutelspacher [Beutelspacher, 1994] proposed a similar illustration using an asymmetric bolt. These objects are assumed to be difficult to copy, in other words it is difficult to forge a signature of somebody else.*

*Following the invention of public key encryption, researchers proposed various ways of signing electronic documents. The key principle is illustrated in Figure 23: a secret signature key plays the role of a seal. Then, given a signed*

Figure 22: Seal

*message, everyone can verify that the signature has been generated by the owner of the secret key (the "seal") using his public key.*
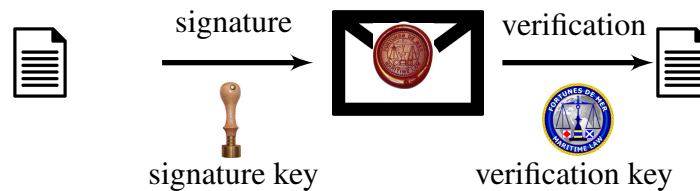


Figure 23: Signature

Using potatoes and ink, kids can make their own seal and "sign" messages they exchange using envelopes. For high school students, one can present RSA signatures [Rivest et al., 1978b] or Elgamal signatures [ElGamal, 1985].

RSA signatures [Rivest et al., 1978b] can also be used as blind signatures. The idea of a blind signature is to be able to sign an encrypted message, and then remove the encryption to obtain a signature on the plaintext message. This is for example used in electronic elections: before voting, each voter asks the authority to verify that they are registered, and to sign their encrypted (in order to preserve the secret of the vote) ballot. Then the voter decrypts the message, and obtains a ballot of his choice, signed by the authority. The voter uses this signed ballot to vote during the election, where the signature proves that he is registered and has the right to vote.

**Illustration 15** (Blind Signature). *As proposed in [Chaum, 1985], blind signatures can be illustrated as in Figure 24: a voter sends an envelope (representing the encryption) that contains his ballot and a sheet of carbon paper. Then the authority signs the envelope on the outside, either using a pen or using his seal, and the voter can open the envelope to obtain the signature on his initial ballot.*
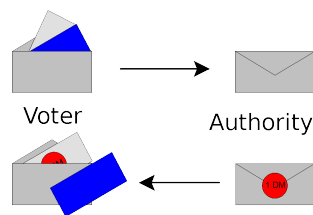
Figure 24: Blind Signature.

This protocol can easily be realized with kids, one only needs some standard office material.

# 7 Illustrating the Diffie-Hellman (DH) Key Exchange with Colors

In 1976, W. Diffie and M. Hellman [Diffie and Hellman, 1976] proposed the first protocol of key exchange. This protocol allows Alice and Bob to establish a shared secret key using a so-called one-way function.

Like the ElGamal encryption, the Diffie-Hellman (DH) key exchange protocol relies on the discrete logarithm problem and works as follows. Alice and Bob know a public number $g$, where $g$ is a public generator of a multiplicative cyclic group. Alice picks a secret random number $a$ and Bob picks a secret random number $b$. Alice sends to Bob $g^a$ then Bob replies $g^b$. Finally, Alice computes the shared key $(g^b)^a$ and Bob can also compute it as $(g^a)^b$.

**Illustration 16** (Diffie-Hellman Key Exchange). *The Diffie-Hellman protocol can be illustrated using colors as follows*[14]. *The illustration, first proposed by Simon Singh [Singh, 1999], relies on two properties of colors:*

- *It is easy to mix two colors, for instance mixing yellow and blue gives green.*

- *Given a color that was obtained by mixing two colors, it is difficult to "separate" the color and obtain the initial colors.*

*For the colored version of the Diffie-Hellman protocol we assume that the color yellow • is a public color, known by everybody. We also assume that the secret color of Alice is the color blue • and that the secret color of Bob is the color red •. Alice and Bob want to agree on a new shared secret color. For this they follow the protocol described in Figure 25, where the name of the white rabbit is Bob.*

---

[14]The following videos show how this protocol works https://www.youtube.com/watch?v=U1kybvKaUeQ or https://www.youtube.com/watch?v=3QnD2c4Xovk
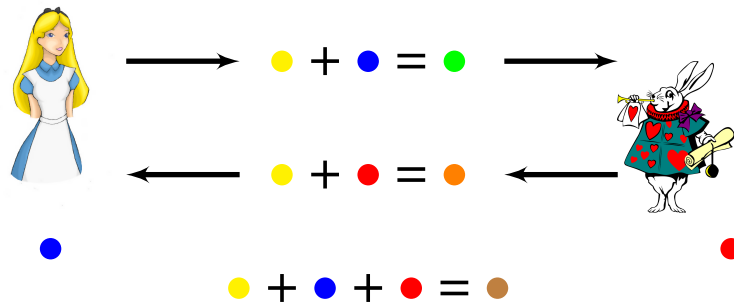
Figure 25: Diffie-Hellman protocol with colors.

*In the first step, Alice sends the color green ● to Bob, which is the result of the public color ● mixed with her secret color ●. Then Bob sends the color orange ● to Alice, which is the result of the public color ● mixed with his secret color ●. Now they both know a common shared secret color brown ● that is computed as follows by Alice and Bob:*

- *Alice mixes the received orange color ● with her secret blue color ●.*

- *Bob mixes the received green color ● with his secret red color ●.*

This protocol can be executed with kids using some transparent paper and colored pens, or even with paint as it is shown in the video. Everyone selects his own color and draws a circle on a transparent paper with their public color and the secret color. Then they can exchange the transparent papers, then they just have to use another secret transparent paper that only contains their secret color to obtain the secret shared color. In practice this protocol might sometimes be difficult to set up, in that case using a drawing software such as GIMP[15] can be a good alternative to convince the students.

# 8 Conclusion

In this paper we proposed child-friendly illustrations of basic modern security concepts that are used in many of the secure electronic devices used by kids today. In the spirit of previous work, we proposed illustrations and analogies that can be used to explain complex security notions to children and non-experts.

As future work, we would like to find similar illustrations for more complex notions like identity based encryption, broadcast encryption, proxy re-encryption or even post-quantum cryptography like lattice based encryption.

---

[15]https://www.gimp.org/

# References

[Arnoux and Finkel, 2010] Arnoux, P. and Finkel, A. (2010). Using mental imagery processes for teaching and research in mathematics and computer science. *International Journal of Mathematical Education in Science and Technology*, 41(2):229–242.

[Baigneres et al., 2010] Baigneres, T., Junod, P., Lu, Y., Monnerat, J., and Vaudenay, S. (2010). *A Classical Introduction to Cryptography Exercise Book*. Springer Publishing Company, Incorporated, 1st edition.

[Bauer, 1997] Bauer, F. L. (1997). *Decrypted Secrets*. Springer Verlag, Berlin.

[Beaver, 2009] Beaver, C. (2009). Cryptology in the classroom: Analyzing a zero-knowledge protocol. *Cryptologia*, 33(1):16–23.

[Bell et al., 2003] Bell, T., Thimbleby, H., Fellows, M., Witten, I., Koblitz, N., and Powell, M. (2003). Explaining cryptographic systems. *Comput. Educ.*, 40(3):199–215.

[Bellare et al., 2001] Bellare, M., Boldyreva, A., Desai, A., and Pointcheval, D. (2001). Key-Privacy in Public-Key Encryption. In Boyd, C., editor, *Advances in Cryptology - Proceedings of ASIACRYPT '01*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582, Gold Coast, Australia. Springer.

[Bellare et al., 1997] Bellare, M., Desai, A., Jokipii, E., and Rogaway, P. (1997). A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 394–403. IEEE Computer Society.

[Bellare et al., 1998] Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P. (1998). Relations among notions of security for public-key encryption schemes. In *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '98, pages 26–45, London, UK, UK. Springer-Verlag.

[Bellovin, 2011] Bellovin, S. M. (2011). Frank Miller: Inventor of the one-time pad. *Cryptologia*, 35(3):203–222. An earlier version is available as technical report CUCS-009-11.

[Beutelspacher, 1994] Beutelspacher, A. (1994). *Cryptology*. Mathematical Association of America, Washington DC.

[Borovik, 2002] Borovik, A. (2002). Implementation of the kid krypto concept. In *MSOR Connections*, 2, pages 23–25. `http://www.maths.manchester.ac.uk/~avb/pdf/crypto.pdf`.

[Bultel et al., 2016] Bultel, X., Dreier, J., Dumas, J.-G., and Lafourcade, P. (2016). Physical zero-knowledge proofs for akari, kakuro, kenken and takuzu. In *Proceedings of the 8th International Conference on Fun with Algorithms*, FUN'16, Berlin, Heidelberg. Springer-Verlag.

[Caballero-Gil and Bruno-Castaeda, 2007] Caballero-Gil, P. and Bruno-Castaeda, C. (2007). A cryptological way of teaching mathematics. *Teaching Mathematics and its Applications*, 26(1):2–16.

[Chaum, 1985] Chaum, D. (1985). Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044.

[Diffie and Hellman, 1976] Diffie, W. and Hellman, M. (1976). New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654.

[Dolev et al., 2000] Dolev, D., Dwork, C., and Naor, M. (2000). Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437.

[ElGamal, 1985] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472.

[Fellows and Koblitz, 1993] Fellows, M. and Koblitz, N. (1993). Kid krypto. In Brickell, E. F., editor, *Advances in Cryptology — CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings*, pages 371–389, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Fellows and Koblitz, 1994] Fellows, M. R. and Koblitz, N. (1994). Combinatorially based cryptography for children (and adults). In *Congressus Numerantium. A Conference Journal on Numerical Themes*, volume 99, page 941.

[Fiat and Shamir, 1987] Fiat, A. and Shamir, A. (1987). How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 186–194, London, UK, UK. Springer-Verlag.

[Forišek and Steinová, 2012] Forišek, M. and Steinová, M. (2012). Metaphors and analogies for teaching algorithms. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, SIGCSE '12, pages 15–20, New York, NY, USA. ACM.

[Fousse et al., 2011] Fousse, L., Lafourcade, P., and Alnuaimi, M. (2011). Benaloh's dense probabilistic encryption revisited. In Nitaj, A. and Pointcheval, D., editors, *Progress in Cryptology - AFRICACRYPT 2011 - 4th International Conference on Cryptology in Africa, Dakar, Senegal, July 5-7, 2011. Proceedings*, volume 6737 of *Lecture Notes in Computer Science*, pages 348–362. Springer.

[Gentry, 2009] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In Mitzenmacher, M., editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM.

[Gentry, 2010] Gentry, C. (2010). Computing arbitrary functions of encrypted data. *Commun. ACM*, 53(3):97–105.

[Goldwasser and Micali, 1984] Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299.

[Gradwohl et al., 2007] Gradwohl, R., Naor, M., Pinkas, B., and Rothblum, G. N. (2007). Cryptographic and physical zero-knowledge proof systems for solutions of sudoku puzzles. In *Proceedings of the 4th International Conference on Fun with Algorithms*, FUN'07, pages 166–182, Berlin, Heidelberg. Springer-Verlag.

[Green, 2014] Green, M. (2014). Zero knowledge proofs: An illustrated primer. `https://blog.cryptographyengineering.com/2014/11/27/zero-knowledge-proofs-illustrated-primer/`.

[Khovanova, 2010] Khovanova, T. (2010). One-way functions. `http://blog.tanyakhovanova.com/2010/11/one-way-functions/`.

[Koblitz, 1997] Koblitz, N. (1997). Cryptography as a teaching tool. *Cryptologia*, 21(4).

[Moran and Naor, 2010] Moran, T. and Naor, M. (2010). Basing cryptographic protocols on tamper-evident seals. *Theor. Comput. Sci.*, 411(10):1283–1310.

[Naccache and Stern, 1998] Naccache, D. and Stern, J. (1998). A new public key cryptosystem based on higher residues. In Gong, L. and Reiter, M. K., editors, *CCS '98, Proceedings of the 5th ACM Conference on Computer and Communications Security, San Francisco, CA, USA, November 3-5, 1998.*, pages 59–66. ACM.

[Naor et al., 1999] Naor, M., Naor, Y., and Reingold, O. (1999). Applied kid cryptography or how to convince your children you are not cheating. In *In Proc. of Eurocrypt 94*, pages 1–12.

[Naor and Yung, 1990] Naor, M. and Yung, M. (1990). Public-key cryptosystems provably secure against chosen ciphertext attacks. In Ortiz, H., editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 427–437. ACM.

[Paillier, 1999] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In Stern, J., editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer.

[Phan, 2005] Phan, R. C.-W. (2005). How to explain block cipher cryptanalysis to your kids. *Cryptologia*, 29(2):148–158.

[Quisquater et al., 1989] Quisquater, J.-J., Guillou, L., Annick, M., and Berson, T. (1989). How to explain zero-knowledge protocols to your children. In *Proceedings on Advances in Cryptology*, CRYPTO '89, pages 628–631, New York, NY, USA. Springer-Verlag New York, Inc.

[Rackoff and Simon, 1993] Rackoff, C. and Simon, D. R. (1993). Cryptographic defense against traffic analysis. In Kosaraju, S. R., Johnson, D. S., and Aggarwal, A., editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 672–681. ACM.

[Rivest et al., 1978a] Rivest, R. L., Adleman, L., and Dertouzos, M. L. (1978a). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180.

[Rivest et al., 1978b] Rivest, R. L., Shamir, A., and Adleman, L. (1978b). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126.

[Rogaway and Shrimpton, 2004] Rogaway, P. and Shrimpton, T. (2004). Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Roy, B. and Meier, W., editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer Berlin Heidelberg.

[Schneier, 1996] Schneier, B. (1996). *Applied Cryptography*. John Wiley & Sons, New York, second edition.

[Shamir et al., 1981] Shamir, A., Rivest, R. L., and Adleman, L. M. (1981). Mental poker. In Klarner, D. A., editor, *The Mathematical Gardner*, pages 37–43. Springer US, Boston, MA.

[Singh, 1999] Singh, S. (1999). *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, New York, NY, USA, 1st edition.

[van Dijk et al., 2010] van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In Gilbert, H., editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer Berlin Heidelberg.

[Vaudenay, 2005] Vaudenay, S. (2005). *A Classical Introduction to Cryptography: Applications for Communications Security*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.