Fabio Rogério Piva

"Addressing human factors in the design of cryptographic solutions: a two-case study in item validation and authentication"

"*Abordando fatores humanos no projeto de soluções criptográficas: dois estudos de caso em validação de itens e autenticação*"

CAMPINAS
2014

University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Fabio Rogério Piva

"Addressing human factors in the design of cryptographic solutions: a two-case study in item validation and authentication"

Supervisor: *Orientador(a):* **Prof. Dr. Ricardo Dahab**

"Abordando fatores humanos no projeto de soluções criptográficas: dois estudos de caso em validação de itens e autenticação"

PhD Thesis presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a PhD degree in Computer Science.

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Ciência da Computação.

This volume corresponds to the final version of the Thesis defended by Fabio Rogério Piva, under the supervision of Prof. Dr. Ricardo Dahab.

Este exemplar corresponde à versão final da Tese defendida por Fabio Rogério Piva, sob orientação de Prof. Dr. Ricardo Dahab.

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

P688a

Piva, Fabio Rogério, 1982-
Addressing human factors in the design of cryptographic solutions : a two-case study in item validation and authentication / Fabio Rogério Piva. – Campinas, SP : [s.n.], 2014.

Orientador: Ricardo Dahab.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Criptografia. 2. Comércio eletrônico. 3. Tecnologia da informação - Aspectos sociais. 4. Redes de computadores - Protocolos. I. Dahab, Ricardo,1957-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Abordando fatores humanos no projeto de soluções criptográficas : dois estudos de caso em validação de itens e autenticação
Palavras-chave em inglês:
Cryptography
Electronic commerce
Information technology - Social aspects
Computer network protocols
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Ricardo Dahab [Orientador]
Ruy José Guerra Barreto de Queiroz
Mylène Christine Queiroz de Farias
Julio César López Hernández
Diego de Freitas Aranha
Data de defesa: 28-03-2014
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Defesa de Tese de Doutorado em Ciência da Computação, apresentada pelo(a) Doutorando(a) **Fabio Rogério Piva**, aprovado(a) em **28 de março de 2014**, pela Banca examinadora composta pelos professores doutores:

Prof(ª). Dr(ª). Ruy José Guerra Barreto de Queiroz
Titular

Mylene Christine Queiroz de Farias

Prof(ª). Dr(ª). Mylène Christine Queiroz de Farias
Titular

Prof(ª). Dr(ª). Julio César López Hernández
Titular

Diego de Freitas Aranha

Prof(ª). Dr(ª). Diego de Freitas Aranha
Titular

Ricardo

Prof(ª). Dr(ª). Ricardo Dahab
Presidente

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

Addressing human factors in the design of cryptographic solutions: a two-case study in item validation and authentication

Fabio Rogério Piva[1]

March 28, 2014

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Ricardo Dahab (Supervisor/*Orientador*)
- Prof. Dr. Julio César López Hernández
  Instituto de Computação - UNICAMP
- Prof. Dr. Diego de Freitas Aranha
  Instituto de Computação - UNICAMP
- Prof. Dr. Ruy José Guerra Barreto de Queiroz
  Centro de Informática - UFPE
- Profa. Dra. Mylène Christine Queiroz de Farias
  Depto. Engenharia Elétrica - UnB

[1] Financial support: FAPESP scholarship (process number 2009/02350-3) 2009–2011; CAPES/DAAD scholarship (process number A/10/71405) 2011–2013

Abstract

Designing secure cryptographic solutions from a purely theoretical perspective is not enough to guarantee their success in a realistic scenario. Many times, the assumptions under which these solutions are designed could not be further from real-world necessities. One particular, often-overlooked aspect that may impact how the solution performs after deployment is how the final user interacts with it (i.e., human factors). In this work, we take a deeper look into this issue by analyzing two well known application scenarios from Information Security research: The electronic commerce of digital items and Internet banking.

Fair exchange protocols have been widely studied, but are still not implemented on most e-commerce transactions available. For several types of digital items (e-goods), the current e-commerce business model fails to provide fairness to customers. A critical step in fair exchange is item validation, which still lacks proper attention from researchers. We believe this issue should be addressed in a comprehensive and integrated fashion before fair exchange protocols can be effectively deployed in the marketplace. More generally, we also believe this to be the consequence of ongoing system-oriented security solution design paradigms that are data-centered, as opposed to user-centered, thus leading to methods and techniques that often disregard users' requirements.

We contextualize how, by overlooking the subtleties of the item validation problem, the current model for buying and selling digital items fails to provide guarantees of a successful transaction outcome to customers, thus being unfair by design. We also introduce the concept of Reversible Degradation, a method for enhancing buy-sell transactions concerning digital items that inherently includes the item validation step in the purchase protocol in order to tackle the discussed problems. As a proof of concept, we produce a deliverable instantiation of Reversible Degradation based on systematic error correction codes (SECCs), suitable for multimedia content. This method is also the byproduct of an attempt to include users' requirements into the cryptographic method construction process, an approach that we further develop into a so-called item-aware protocol design.

From a similar perspective, we also propose a novel method for user and transaction authentication for Internet Banking scenarios. The proposed method, which uses Visual

Cryptography, takes both technical and user requirements into account, and is suitable as a secure – yet intuitive – component for practical transaction authentication scenarios.

Resumo

O projeto de soluções criptográficas seguras a partir de uma perspectiva puramente teórica não é suficiente para garantir seu sucesso em cenários realistas. Diversas vezes, as premissas sob as quais estas soluções são propostas não poderiam estar mais longe das necessidades do mundo real. Um aspecto frequentemente esquecido, que pode influenciar em como a solução se sai ao ser integrada, é a forma como o usuário final interage com ela (i.e., fatores humanos). Neste trabalho, estudamos este problema através da análise de dois cenários de aplicação bem conhecidos da pesquisa em Segurança da Informação: O comércio eletrônico de itens digitais e Internet *banking*.

Protocolos de trocas justas tem sido amplamente estudados, mas continuam não sendo implementados na maioria das transações de comércio eletrônico disponíveis. Para diversos tipos de itens digitais (*e-goods*), o modelo de negócios atual para comércio eletrônico falha em garantir justiça aos clientes. A validação de itens é um passo crítico em trocas justas, e recebeu pouca atenção dos pesquisadores. Nós acreditamos que estes problemas devam ser abordados de forma integrada, para que os protocolos de trocas justas possam ser efetivamente implementados no mercado. De forma geral, acreditamos também que isso seja um reflexo de paradigmas de projeto orientado a sistemas para soluções de segurança, que são centrados em dados em vez de usuários, o que resulta em métodos e técnicas que frequentemente desconsideram os requisitos de usuários.

Contextualizamos como, ao subestimar as sutilezas do problema da validação de itens, o modelo atual para compra e venda de itens digitais falha em garantir sucesso, na perspectiva dos compradores, para as transações – sendo, portanto, injusto por definição. Também introduzimos o conceito de Degradação Reversível, um método que inerentemente inclui o passo de validação de itens em transações de compra e venda com a finalidade de mitigar os problemas apresentados. Como prova-de-conceito, produzimos uma implementação de Degradação Reversível baseada em Códigos Corretores de Erros Sistemáticos (SECCs), destinada a conteúdo multimídia. Este método é também o subproduto de uma tentativa de incluir os requisitos do usuário no processo de construção de métodos criptográficos, uma abordagem que, em seguida, evoluímos para o denominado projeto de protocolos orientado a itens.

De uma perspectiva semelhante, também propomos um método inovador para a autenticação de usuários e de transações para cenários de Internet *banking*. O método proposto, baseado em Criptografia Visual, leva em conta tanto requisitos técnicos quanto de usuário, e cabe como um componente seguro – e intuitivo – para cenários práticos de autenticação de transações.

The following people played key roles in the development of this thesis:

- *My adviser and mentor, Prof. Dr. Ricardo Dahab, for being always understanding with my arguably chaotic ways.*
- *My estimated colleague Dr. Bernd Borchert, for challenging my ideas and motivating me to always push just a little bit further.*
- *My girlfriend Natalyia Popova, for showing me that moving your whole life across the ocean is not such a big deal – if your heart is in the right place.*
- *My parents, Francisco Carlos Piva and Vânia Aparecida Piva, for being there with me every single time that I had to start from scratch.*

Acknowledgements

We thank the financial support of the São Paulo Research Foundation (FAPESP), the National Council for Scientific and Technological Development (CNPq) and the German Academic Exchange Service (DAAD). We also thank the Center for Advanced Security Research Darmstadt (CASED) for hosting the first author during his leave from the University of Campinas (UNICAMP).

"Eu prefiro ser essa metamorfose
ambulante, do que ter aquela velha
opinião formada sobre tudo. "
Raul Seixas

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introduction

1.1 A tale of two cities: has Information Security failed us?

Over the past ten years, selling digital content has become an attractive business, mainly due to the fast increase of consumers' interest in that kind of media. The latest technological trends – such as cheaper and faster broadband internet connections, greater storage space, and devices that provide users with easier access to their data on-the-go – contributed to the development of a solid market for these so-called e-goods. And although several virtual retailers have emerged in order to supply this demand for digital content, most of them have chosen to adopt a real-world-based business model that, however successful for selling physical products, is unsuitable for trading digital goods; it has been observed that physical and digital items are intrinsically too different to be negotiated in the same way [19, 63].

In the classical real-world commercial model, where transactions concern the purchase and sale of physical items, a customer is able to return a product if it does not satisfy his initial expectations – for instance, in cases where the advertised description of that product does not satisfy its actual features or when a different product is delivered by the seller due to some bureaucratic/technical mistake. This is usually true even for e-commerce retailers, much like what happens when the product is purchased at a physical store. However, when the product of interest is a digital item – such as an audio file or digital picture purchased online, for instance – return policies usually do not apply [8], mainly because it is trivial for the buyer to create an identical copy of a digital item that has already been obtained. In such cases, unsatisfied customers can do little to recover their money, or even exchange the delivered product by another one – even in situations where the inaccurate product was mistakenly delivered instead of the desired one [63].

Another online service commonly engaged by the everyday user is Internet banking (and, more recently, mobile banking). In this scenario, a user is first required to authenticate himself to a remote bank server, to which he will then request some particular task to be performed (such as a money transfer). Other security requirements in this scenario include the guarantee that the user is, in fact, communicating with the bank (i.e., counterpart authentication for the user), and the guarantee that the transaction parameters transmitted to the bank by the user have not been tampered with by an adversary during end-to-end transmission (i.e., transaction authentication).

Even though several solutions have been proposed for authentication scenarios – and specifically, for Internet banking (see Chapter 5) – the Man-in-the-Middle (MitM) attack continues to elude researchers and service providers alike. In fact, and regardless of being extensively researched in the past, the MitM attack remains a real problem in practice [26, 41, 33, 32, 56, 21] – mostly due to the fact that previous solutions have often been proposed under unrealistic assumptions that do not take into account users' needs or average behavior.

We believe that these two apparently unrelated issues share a common foundation: they are addressed from an essentially flawed design perspective, that fails both to appropriately model the underlying cause of each problem and to take human factors into account – either as an obstacle to security, or as a tool to provide it. For the remainder of this work, we further extend this argument and approach each individual scenario from a human-centered perspective towards the design of cryptographic solutions.

1.2 Information security vs. user perception of security: a brief perspective

Cryptographic artifacts[1] are almost entirely designed with the sole purpose of protecting **information** (as opposed to **users** or **parties**) from unauthorized actions performed by unauthorized agents. While we do acknowledge the importance of securing information, we also notice that even when information is successfully protected within a secure system, the achieved security is not necessarily perceived by its users. This *perception of security* is, therefore, more associated to the notion of *trust* [22, 36] and less to the notion of *security* itself – and plays a major role on user adoption of new systems.

In the context of electronic commerce, for instance, the role of trust has been extensively researched [22, 36, 18, 78, 77, 76]. Particular instances of fair exchange protocols [10], for instance, rely on trusted third parties (TTPs) as intermediate entities for

[1] We shall refer as *cryptographic artifacts* to any cryptographic software (encryption algorithms, hash functions, etc) and/or hardware (hardware tokens, cryptographic co-processors, etc).

sensitive transactions – an attempt to increase user trust in the system through the presence of a previously-attested trustworthy entity. In fact, it has been said that the lack of trust is the main reason why many consumers and companies choose not to engage in e-commerce [22], a conclusion that we share. It seems that many users still tend to be reluctant in engaging in online financial activities, even when proper security measures are available and implemented – for the simple reason that they do not trust their transaction counterparts, the communication channel, or the security measures themselves.

One possible explanation for this could lie on the very nature of those measures: they are usually based on complex mathematical problems that do what they are supposed to do with no (or very little) user interference [73, 90, 25, 71, 4]. For most users, it never becomes clear how these artifacts actually protect their information or interests – and therefore their perception of security is often very limited. In that sense, we believe that cryptographic artifacts should be designed, whenever possible, with the purpose of providing not only security regarding sensitive information, but also perception of security for their users. That belief is supported by the fact that user experience should be a relevant factor on the design process of every computational system meant to be used “by people”; when cryptographic artifacts – which have as main objects pieces of sensitive information – are concerned, we believe that user experience can be translated into perception of security.

1.3 Objectives

This thesis’ main goal is to approach the design of cryptographic solutions while accounting for human factors, as well as evaluating how those factors can be used as assets for guaranteeing security requirements. For that matter, the following specific goals are devised:

1. the study, from a human factors-oriented perspective, of the item validation problem and its impacts in real-world applications in which fairness is required;

2. a survey on special properties of digital items and discussion of their impacts on fair exchange protocols;

3. the proposal of an item validation framework for enabling the fair exchange of particularly hard-to-handle digital items;

4. the implementation, as a proof-of-concept, of an instantiation of the above framework;

5. the study, from a human factors-oriented perspective, of the Internet banking authentication problem and the Man-in-the-Middle attack;

6. the proposal of an authentication solution that addresses the Man-in-the-Middle attack under realistic assumptions;

7. the implementation, as a proof-of-concept, of the above authentication method.

1.4 Thesis contributions

In this section we summarize the contributions of this Ph.D. research project.

1.4.1 Main results

- The proposal of the item validation problem as a relevant topic of research in fair exchange e-commerce literature (remainder of Chapter 1);

- a survey on most commonly observed properties of digital items found in related literature (Chapter 2);

- the proposal of perception of security (additionally to security of information) as a relevant aspect for security method design – specially in user-oriented applications;

- the proposal of the reversible degradation concept as a model for enabling item validation of certain items in fair exchange protocols (Chapter 3);

- a proof-of-concept implementation of a reversible degradation instance (Chapter 3);

- a discussion of the generic item approach to fair exchange protocol design and proposal of a non-generic, item-aware approach to the process (Chapter 4);

- the proposal and proof-of-concept implementation of a transaction authentication method based on Visual Cryptography (Chapter 5), which effectively prevents the Man-in-the-Middle attack.

1.4.2 Publications

- Full paper (first author): "*Modern fair exchange protocol design: dealing with complex digital items.*", In: XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2013, Manaus/Brazil.

- Journal article (first author): "*E-commerce of digital items and the problem of item validation: introducing the concept of reversible degradation.*", In: Journal of Applicable Algebra in Engineering, Communication and Computing, 2013. Springer-Verlag Berlin Heidelberg.

- Extended abstract (first author): "*Using systematic error correcting codes for reversible degradation of multimedia content.*", In: 18th International Conference on Applications of Computer Algebra (ACA), 2012. Sofia/Bulgaria.

- Full paper (first author): "*E-commerce and fair exchange - The problem of item validation.*", In: International Conference on Security and Cryptography (SECRYPT), 2011. Seville/Spain.

- Technical report (first author): "*E-commerce and fair exchange: the problem of item validation.*", Institute of Computing, University of Campinas, 2011. Campinas/Brazil.

1.4.3 Submitted patent requests

- Degradação Reversível: um método para validação segura de itens digitais em protocolos de comércio eletrônico;

- Método de Criptografia Visual para compartilhamento de segredos múltiplos com transparência reutilizável.

1.5 Document outline

The remainder of this document is organized as follows: Chapter 2 describes the scenario for our first case of study (namely the unsuitability of the current e-commerce model to today's consumer's and seller's needs) and presents relevant previous results on this subject. Chapter 3 presents our concept of reversible degradation, an abstraction for solving the problem of fairly exchanging, validating and selling multimedia content over the Internet. These results provide us with a new perspective towards the very process of designing fair exchange protocols, which we present in Chapter 4. Chapter 5 contextualizes our second case of study (namely transaction authentication applications, which we approach from an Internet Banking perspective) and presents our proposed solution – which, by relying on Visual Cryptography, takes advantage of human factors for security purposes. We conclude in Chapter 6 with some final remarks on how human factors should be not only addressed, but also taken advantage of in the design of cryptographic solutions, and what future research on this topic should focus on.

Capítulo 2

Item validation: Motivation and related research

In this chapter we look into the item validation problem of fair exchange protocols, and how it affects the current model for buying and selling digital products in electronic commerce applications. We also review well-known aspects and issues of fair exchange protocols research [10, 67, 6, 58, 82, 63, 19] that further strengthen the claim that the current model is unsuitable to promote long-term customer satisfaction. This approach allows us to isolate the problem to the context of fair exchange and provides us with the required know-how for further proposing the reversible degradation concept (which we will discuss in Chapter 3) as a suitable enhancement to selling protocols in order to inherently enforce item validation.

2.1 Fair exchange protocols and e-commerce

Fair exchange protocols were proposed by Asokan [10] as a solution to the problem of two mutually distrusting parties interested in exchanging digital items atomically. Many variations of Asokan's original protocols have since been studied [67, 6, 58], but most of them were too cumbersome or required too many resources to be considered practical for real applications. Probabilistic fair exchange [46], for instance, required a great number of messages to be transmitted, in order to achieve only a probabilistic, more relaxed form of fairness.

Arguably, the most successful instances of fair exchange in the real world are optimistic fair exchange protocols [82, 12, 75] – two-party protocols that rely on a mutually trusted third party (further referred to as TTP or trustee) to handle exceptions that may arise during the exchange. These protocols quickly became the focus of fair exchange research, and were used as the core of several pioneering fair-exchange-based e-commerce projects [53, 40].

Much work has been dedicated to the formalization of fair exchange protocol design and its subtleties. Gärtner et. al. [30] give the first steps towards a formal definition of fairness; other authors follow similar approaches to different fair exchange properties, such as non-repudiation [38] and timeliness [60]. As for verification methods, very few of them seem suitable to the analysis of optimistic fair exchange protocols, mostly due to their multi-protocol nature. Some recent works accomplished interesting results [65, 64, 62] with the adaptation of the Strand Spaces method [79] for supporting optimistic protocols.

However, most current e-commerce stores do not implement fair exchange in their business models; a simple web search reveals several Apple's iTunes Store user complaints about mistaken music files being purchased due to inaccurate description of the products; also, the Digital Downloads section on Amazon.com contains several customer comments on the same subject. To worsen the problem, most companies openly adopt a no-refund policy when it comes to selling digital products – even in the case of a mistaken purchase [8].

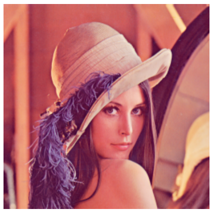
Such problems relate to an essential, but not sufficiently explored, aspect of fair exchange protocols: the **item validation** step. The original definition of fairness states that "*an exchange is fair if at the end of the exchange, either each player receives the item it expects or neither player receives any additional information about the other's item*" [10]. For that end, aside from ensuring the atomicity of the exchange, the protocol must specify when and how a party can check whether the item she has just received (or is about to receive) is the one she desires. This is, however, a delicate process that may be influenced by the characteristics of the items being exchanged, by the available resources and by the structure of the protocol itself [63].

As such, we consider the problem of item validation of digital items a very relevant topic of research, and that e-commerce concerning digital items would greatly benefit from a better understanding of its subtleties. We also believe that the lack of attention to this issue is the very reason why fair exchange protocols are not yet widely implemented in the current e-commerce business models. In this chapter, we further analyze how the currently implemented business model for buying and selling digital items (specifically multimedia content) allows unfair outcomes, and how this issue undermines customer trust in online purchases as a whole.

Since we approach these inherent issues in a top-down fashion (i.e., by initially taking into account the structural and semantic nature of the items that should be exchanged in the protocol), we also provide a set of guidelines that can assist the modern fair exchange protocol designer to build realistic solutions for real-world applications. We call this design approach **item-aware protocol design** [61], as opposed to the *generic item protocol design* approach introduced in Asokan's original work and followed by many authors after him. As far as generic item protocol design is concerned, the exchanged items are seen as generic bit streams with few or no particular properties of relevance to protocol design. We believe, however, that in most current contexts, items do have inherent complexity that may interfere with transactions, and exhibit characteristics that either make the exchange easier, or become obstacles for enforcing successful (fair) outcomes. Those properties are usually left aside during protocol design for the sake of “simplicity”of explanation, which often results in proposals that are inaccurate, inefficient or inadequate [49, 87, 89, 59] for most real-world applications – where items are far from generic bit streams. To the extent of our knowledge, only a few works have taken into account how the nature of the exchanged items may impact transaction outcomes [82, 19, 60, 63, 61].

2.2 Current model description

The currently adopted model for electronic commerce of digital content is the following: First, the buyer registers with the seller, thus obtaining an account through which the transaction will be carried out. Then, while logged in the store’s system (which might be a website or client software, for instance), the buyer chooses the product he desires to purchase; he must check carefully whatever descriptions – such as feature lists, pictures, or samples – are available for that product. Figure 2.1 illustrates a hypothetical digital item for sale and its possible description.



(a) Item i

- *Item Summary:* Portrait of model Lena Söderberg
- *Keywords:* hat, bust, plumes, portrait
- *File Specs:* PNG image (bitmap), RGB, 256x256 resolution

(b) Description $desc(i)$ of i

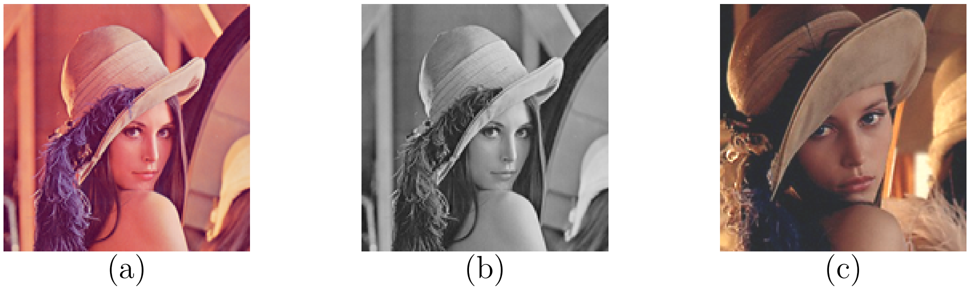
Figura 2.1: Desired item i and its description as a list of specifications.

In this model, the item is not publicly available for the interested customer before the purchase transaction is completed (otherwise he would be able to acquire it without paying for it). Instead, only the description, which in this example is the list of features ilustrated in Figure 2.1b, is available before payment.

Once satisfied with this description, the buyer places an order for that particular product – he now has a mental image of what he expects to receive. He must then pay for it; this may happen either by revealing his credit card number to the store, or by depositing money to an external account and then informing the seller, for instance. Either way, upon receiving the payment, the seller finally releases the product to the buyer – maybe by sending him a temporary link for downloading it or as an attachment in an email message.

If the buyer pays but never receives the product, dispute might be started. Most buyers would first try to contact the store, which will generally solve the problem to avoid bad reputation. Mostly, in the case of a digital file's purchase, the store would simply resend the item to the buyer, without losing any money. This is only the case because of the **idempotency** [10] property of e-goods: if a bit stream was to be received by a user, it wouldn't make a difference if that same bit stream was received multiple times, since this would mean that several exact copies of the same e-good was being received. One should notice that, for physical goods, this would not be the case: if the store was required to re-send a physical item, this would represent loss of money to the seller, and a malicious buyer would be able to retain two or more identical – but several, nevertheless – instances of a product.

However, a completely different situation occurs if the buyer receives the product, but is not satisfied with it – which may happen if the description about a certain aspect of the product had been left vague or ambiguous by the store. In such cases, the buyer might find himself in an unfair situation – having paid for a product that does not meet his expectations. Figures 2.2a and 2.2b show two candidates for delivery that match the description shown in Figure 2.1b. One could point that the problem could, in this instance, be easily solved by adding the word "color" to the description – but then Figure 2.2(c) would still be a candidate for delivery.

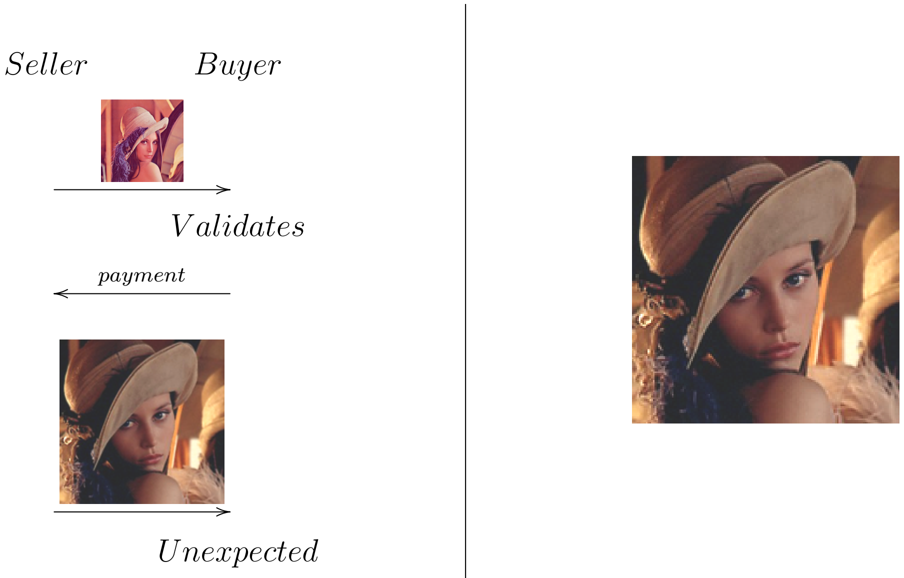


(a) (b) (c)

Figura 2.2: Three different files that show pictures of the model Lena Söderberg. Figures (a) and (b) equally satisfy any description that does not mention color properties, and Figures (a) and (c) could be mistaken even if color is mentioned – which could lead to the wrong file being delivered.

In cases of mistaken delivery, both the store and any external judge might refuse to intervene in favor of the buyer, for the current model assumes that it is his responsibility to carefully check the product description before paying for it. The fact that the buyer is not able to return the wrong item – without possibly keeping a copy for himself – in exchange for the right one, makes the store unable to distinguish a genuine mistake from a pay-for-one-and-take-two con. We refer to this issue as the **unreturnability** property of digital items, and believe that it is one of the reasons why e-goods must be traded differently than physical products.

Unfortunately, if the digital product in question is also an **indescribable item** [19, 63], there can be absolutely no guarantee to a buyer about the outcome of the purchase, in this model. Even if a sample of the file is used as description – a reduced thumbnail of an image file, or a lower bit rate fraction of a song file might be available for download beforehand, for instance – the above mentioned problem might still occur, as we shall see in Section 4.3. In fact, most of the times, the buyer has no guarantee that the item corresponding to the sample is in fact the one to be delivered; as shown in Figure 2.3a, he might very well download a thumbnail sample of image file i, engage in a transaction for acquiring the larger version of it, and end up with a copy of image file i' illustrated in Figure 2.3b – possibly due to some internal error in the store system, or even human error when advertising the item.



(a) Buyer desires i from Figure 2.1a (b) Item $i' \neq i$ which also satisfies $desc(i)$

Figura 2.3: The current model allows a buyer to pay for a file and receive a different one.

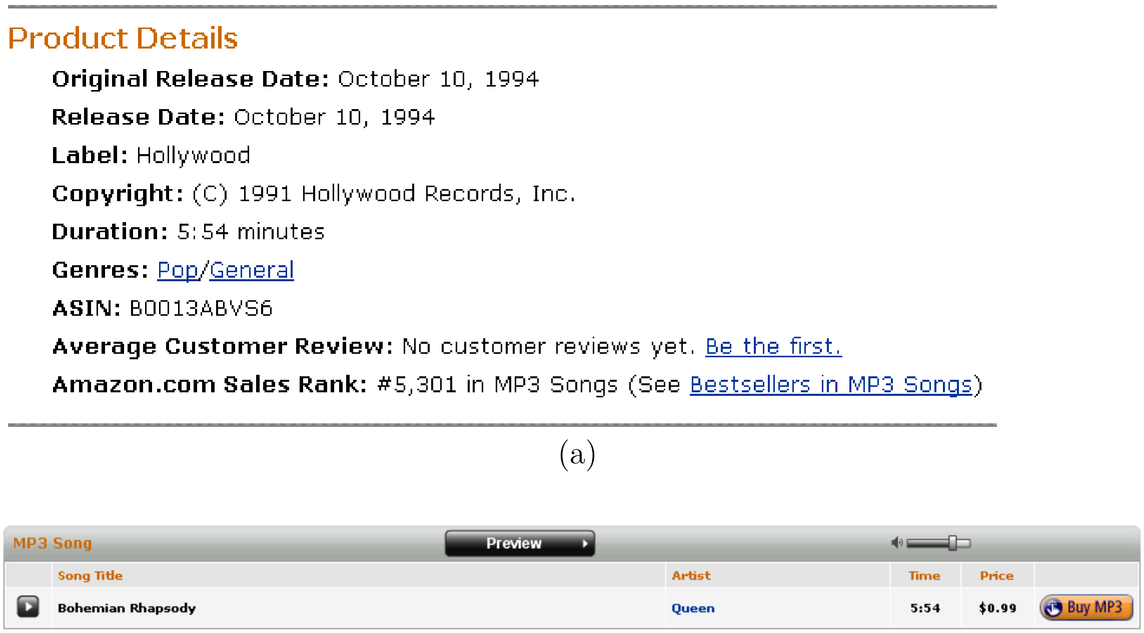
Dispute resolution for this kind of situation would be rather difficult, since the buyer would not be able to return file i' in exchange of desired file i; the store might rightfully allege that the buyer would be able to retain a copy of file i' for himself, which would leave it in an unfair situation. One should notice that even if the description from Figure 2.1b was used instead of the thumbnail to validate the transaction, the wrong item i' would still be a candidate for delivery.

Situations like this happen more often than one might imagine. In the next section we present a few examples of unsatisfied customers, gathered from communities of users of one of the most proeminent e-stores available – namely Amazon.com.

2.3 Real examples of unfairness in e-commerce

In this section we present a few examples of unsatisfied consumers of digital music. The fact that digital music files are examples of indescribable items (as we shall better define in Section 2.5) makes them very difficult to validate and hence exchange fairly. Without a good description, item validation becomes difficult and error prone – because the buyer is not able to accurately decide whether the item he is about to pay for is the one he desires, or not.

Currently, one of the most popular services for buying songs remains Amazon MP3 Downloads, which follows the business model described in Section 2.2. However, in order to strengthen item validation, the store also provides buyers not only with a textual description of products, but also with a limited preview of thirty seconds of each song offered in their catalog, as illustrated in Figure 2.4.



Product Details

Original Release Date: October 10, 1994

Release Date: October 10, 1994

Label: Hollywood

Copyright: (C) 1991 Hollywood Records, Inc.

Duration: 5:54 minutes

Genres: Pop/General

ASIN: B0013ABVS6

Average Customer Review: No customer reviews yet. Be the first.

Amazon.com Sales Rank: #5,301 in MP3 Songs (See Bestsellers in MP3 Songs)

(a)

MP3 Song

	Song Title	Artist	Time	Price	
▶	Bohemian Rhapsody	Queen	5:54	$0.99	Buy MP3

(b)

Figura 2.4: Amazon MP3 Downloads description of an item. Figure (a) shows a list of details about the file, while Figure (b) shows a limited preview button.

One might think that, with the addition of a preview of the song to the description, the chances of someone buying the wrong file would be negligible. However, as Figure 2.5 shows, this may not always be true.

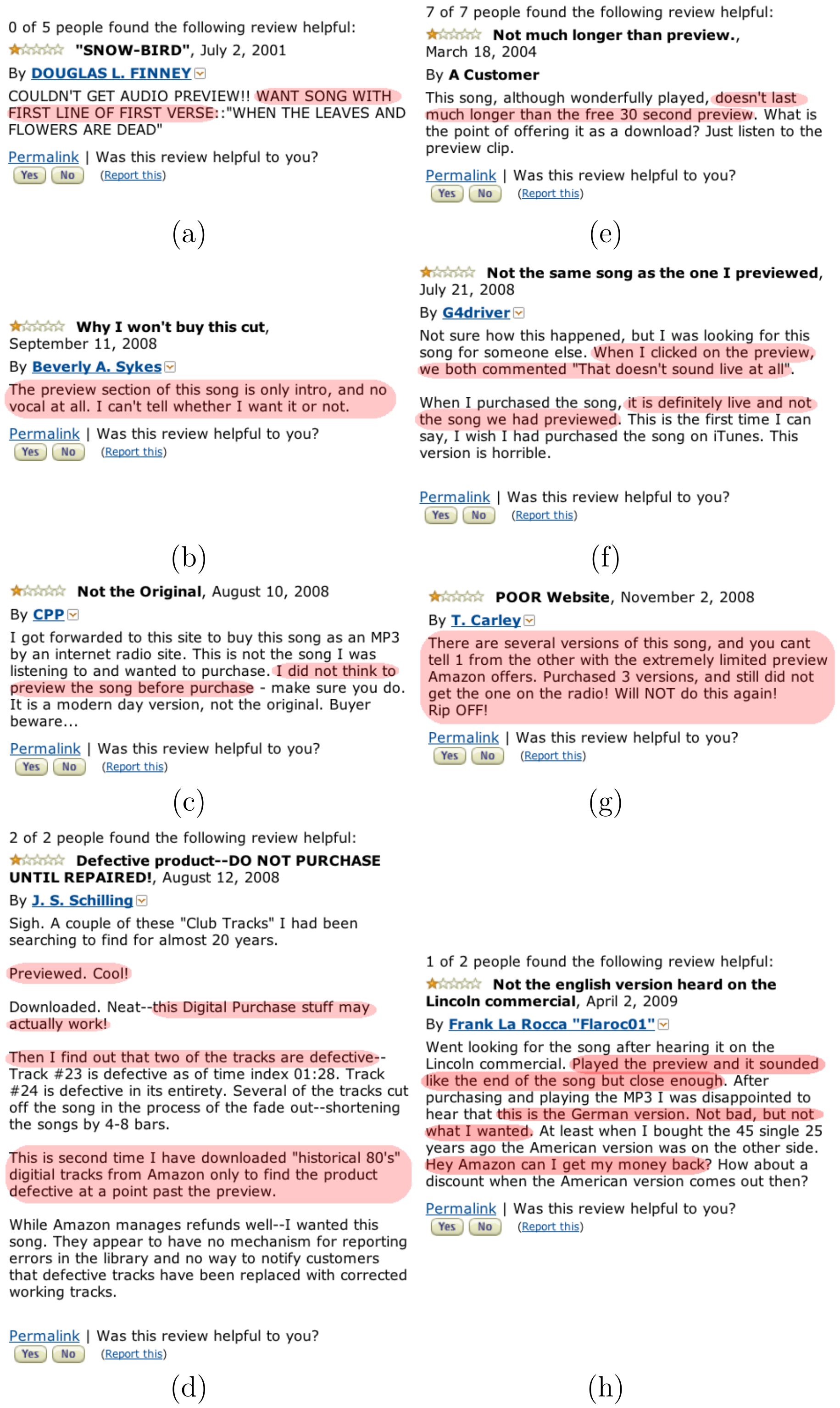
As we previously stated, the current business model for multimedia, as well as several other special items, is anything but fair. In the following sections we relate this issue to the problem of item validation and other fair exchange subtleties, as well as suggest the concept of reversible degradation as a fair alternative model to the current e-commerce paradigm for digital items.

2.4 The problem of item validation

Optimistic fair exchange protocols [10] usually follow a common sequence of events: Let us suppose that two parties P and Q are willing to exchange two generic digital items i_P and i_Q through an asynchronous channel. A common requirement is that P and Q know beforehand the descriptions $desc(i_Q)$ and $desc(i_P)$, respectively, of their expected items; there must also be a publicly available function $validate(i, d)$ which takes an item i and a description d and returns $TRUE$, if d accurately describes i, or $FALSE$ otherwise.

The parties then engage in the fair exchange protocol, initially gathering sufficient information to prove the validity of the transaction (known as the non-repudiation requirement of fair exchange protocol design); this step is crucial to allow for internal or external dispute resolution, in case of exceptions. After that, they proceed to the exchange of the items of interest. When a party receives an item – which can be presented in an encrypted form [14] to that party – it executes $validate()$ using the received item and the known description as parameters, and checks the result to decide whether the item is the expected one. If the function returns $FALSE$, then the protocol must be robust enough to either allow the cheated party to recover the correct item, usually by invoking an offline TTP and providing her information about the unfair transaction, or to stop her counterpart from getting the other item, if it has already been revealed.

Such protocols are called *optimistic* because, by relying on the assumption that in most cases the transaction will be conducted by honest parties, TTP intervention will be limited to the less frequent scenarios in which one of the parties is not satisfied with the transaction outcome. In such cases, this intervention can happen within the protocol run (through the use of subprotocols designed for communication between the unsatisfied party and the TTP) or in a separate event, commonly referred to as *dispute resolution*. In dispute resolution, the TTP acts as a judge – analyzing the transaction data and deciding whether the unsatisfied party's claims are legitimate or not. If the transaction outcome is either ambiguous or subject to interpretation (as we illustrated in Chapter 2), however, this decision can be hard to make.



0 of 5 people found the following review helpful:

★☆☆☆☆ **"SNOW-BIRD"**, July 2, 2001

By **DOUGLAS L. FINNEY**

COULDN'T GET AUDIO PREVIEW!! WANT SONG WITH FIRST LINE OF FIRST VERSE::"WHEN THE LEAVES AND FLOWERS ARE DEAD"

Permalink | Was this review helpful to you?

Yes No (Report this)

(a)

★☆☆☆☆ **Why I won't buy this cut**, September 11, 2008

By **Beverly A. Sykes**

The preview section of this song is only intro, and no vocal at all. I can't tell whether I want it or not.

Permalink | Was this review helpful to you?

Yes No (Report this)

(b)

★☆☆☆☆ **Not the Original**, August 10, 2008

By **CPP**

I got forwarded to this site to buy this song as an MP3 by an internet radio site. This is not the song I was listening to and wanted to purchase. I did not think to preview the song before purchase - make sure you do. It is a modern day version, not the original. Buyer beware...

Permalink | Was this review helpful to you?

Yes No (Report this)

(c)

2 of 2 people found the following review helpful:

★☆☆☆☆ **Defective product--DO NOT PURCHASE UNTIL REPAIRED!**, August 12, 2008

By **J. S. Schilling**

Sigh. A couple of these "Club Tracks" I had been searching to find for almost 20 years.

Previewed. Cool!

Downloaded. Neat--this Digital Purchase stuff may actually work!

Then I find out that two of the tracks are defective--Track #23 is defective as of time index 01:28. Track #24 is defective in its entirety. Several of the tracks cut off the song in the process of the fade out--shortening the songs by 4-8 bars.

This is second time I have downloaded "historical 80's" digitial tracks from Amazon only to find the product defective at a point past the preview.

While Amazon manages refunds well--I wanted this song. They appear to have no mechanism for reporting errors in the library and no way to notify customers that defective tracks have been replaced with corrected working tracks.

Permalink | Was this review helpful to you?

Yes No (Report this)

(d)

7 of 7 people found the following review helpful:

★☆☆☆☆ **Not much longer than preview.**, March 18, 2004

By **A Customer**

This song, although wonderfully played, doesn't last much longer than the free 30 second preview. What is the point of offering it as a download? Just listen to the preview clip.

Permalink | Was this review helpful to you?

Yes No (Report this)

(e)

★☆☆☆☆ **Not the same song as the one I previewed**, July 21, 2008

By **G4driver**

Not sure how this happened, but I was looking for this song for someone else. When I clicked on the preview, we both commented "That doesn't sound live at all".

When I purchased the song, it is definitely live and not the song we had previewed. This is the first time I can say, I wish I had purchased the song on iTunes. This version is horrible.

Permalink | Was this review helpful to you?

Yes No (Report this)

(f)

★☆☆☆☆ **POOR Website**, November 2, 2008

By **T. Carley**

There are several versions of this song, and you cant tell 1 from the other with the extremely limited preview Amazon offers. Purchased 3 versions, and still did not get the one on the radio! Will NOT do this again!
Rip OFF!

Permalink | Was this review helpful to you?

Yes No (Report this)

(g)

1 of 2 people found the following review helpful:

★☆☆☆☆ **Not the english version heard on the Lincoln commercial**, April 2, 2009

By **Frank La Rocca "Flaroc01"**

Went looking for the song after hearing it on the Lincoln commercial. Played the preview and it sounded like the end of the song but close enough. After purchasing and playing the MP3 I was disappointed to hear that this is the German version. Not bad, but not what I wanted. At least when I bought the 45 single 25 years ago the American version was on the other side. Hey Amazon can I get my money back? How about a discount when the American version comes out then?

Permalink | Was this review helpful to you?

Yes No (Report this)

(h)

Figura 2.5: Representative examples of unfair situations occurred due to inaccurate validation of the purchased item.

One should notice that the availability of both the function *validate*() and accurate descriptions of the items are then essential to ensure fairness in this type of protocol. Previous researchers have attempted to approach this problem, for some particular classes of items, by focusing on how the validation artifact (that is, the information made available to the party in the protocol in order to allow her to perform item validation) is constructed. It has been noted that, by embedding the desired item into the validation artifact itself – as opposed to producing a separate validation artifact merely containing partial information about the item – both the item validation step and dispute resolution process become more reliable [14, 54, 15]. We shall further use the term atomic validation artifact to refer to such approaches, as it is also in the core of our own proposed method.

Nevertheless, for some particular items, providing an accurate description (as well as validation artifacts) can be a hard task [19, 63]; we shall return to the issue of item (in)describability in Section 2.5. In fact, even for describable items, validation may be non-trivial. The protocol designer must carefully ponder when in the protocol run the parties will be required to perform validation, and whether the validation artifact will be atomic or non-atomic. Also, should the validation be performed by the parties themselves, or should a TTP be assigned for this task? And if we transfer this responsability to a TTP, how can we describe the *validate()* function so that we can ensure that parties will agree with its evaluation? All those questions must be taken into account when designing a fair exchange protocol.

2.5 Special properties of items (and how they affect validation)

The first proposals for fair exchange protocols [10] regarded items as generic, forwardable sequence of bits, but soon researchers took interest in particular instances of the problem. Some special types of items, such as digital cash, multimedia files etc., may present special properties that should be taken into account during protocol design. In this section we present some of the properties that particular digital items may have and how they affect fair exchange protocol design, as presented by previous researchers.

In this section we look into the items to be exchanged, in light of some specific, commonly-observed properties. As we shall discuss, the level of fairness obtained in fair exchange protocols may highly depend on the characteristics of the items being exchanged themselves, and so these properties should always be taken into account during the process of proposing a new protocol.

2.5.1 Idempotency/Copiability

Perhaps the most relevant difference between digital and physical items is that the first ones are easy to copy. In fact, many fair exchange protocols rely on the fact that receiving a digital item more than once is the same as receiving it once – as digital items are **idempotent** (or **copiable**) [10]. Under that perspective, digital items are indeed essentially sequences of bits, and thus creating a copy of a particular idempotent item makes for an identical copy of that item itself.

Although idempotency can sometimes be an advantage for protocol design – see Section 2.5.3 to see how copies of items can help enforce fairness – the opposite may also be true. For instance, dispute resolution becomes a hard matter when a participant is not able to return an item without possibly retaining a copy for himself – in case of a mistaken delivery, for instance. When physical items are exchanged (as in physical products buying/selling), any mistakes can be easily undone by simply returning the wrong item in exchange for the correct one. It is, therefore, easy to address – by means of return policies – situations in which parties become unsatisfied with the outcome of a particular transaction that concerns physical items.

In exchanges concerning digital items, however, this might not always be the case. Unless the wrong item is *revocable* (see Section 2.5.4 for revocable items), return policies often do not apply [9]. This fact requires that, in order to avoid undesired outcomes, fair exchange protocols must predict and minimize “buying a pig in a poke[1]” scenarios, in which a party is left unsatisfied with the acquired item – which can be particularly hard for *indescribable* items (see Section 2.5.2 for describability).

2.5.2 (In)Describability

Fair exchange protocols typically require that a description of each item must be publicly available (or directly delivered) to parties before the exchange takes place. Such protocols include a critical step – the *item validation* step [63, 61] – in which a party is usually required to check whether the item she has received (or is about to receive) satisfies that description or not.

Such description must, however, be univocal if a party is to be assured about the outcome of the exchange. A univocal description is regarded as a set of characteristics that uniquely define an item – with no other similar item being able to entirely satisfy that particular description. When providing a univocal description of any form for a particular item is possible, we regard it as being **describable**. If only non-univocal descriptions are

[1] “To buy a pig in poke” is an idiom associated to a scenario in which an individual, upon trying to purchase a good-quality pig in a bag, ends up with a low-quality pig because he or she did not carefully check what was in the bag before paying for it – believing the pig’s previous owner’s promises instead.

possible instead, the interested party might be misled into inaccurately validating an item that, while satisfying said description, is nevertheless inherently different than the one she expects to receive. We regard items that can only be described by non-univocal descriptions as being **indescribable** [19, 63, 61].

For an example of this issue, suppose that a party P is willing to obtain a picture i of the model Lena Söderberg, famous for its appearance as case of study in image processing literature. P could be satisfied with a description $desc(i)$ consisting of the following list of words: Lena Söderberg, image processing muse, model, famous, hat, PNG file, face. After engaging counterpart Q in the exchange and delivering her own item i' (possibly some sort of digital payment, for instance), P would expect to receive the file pictured in Figure 2.2(a), but could be surprised by the delivery of Figure 2.2(b) instead. One could notice that the problem could be easily solved by adding the word "color" to the description, but even in that case Figure 2.2(c) would still be a candidate for delivery.

It has been noted that not only pictures, but all forms of multimedia content are naturally **indescribable** – for univocal descriptions for such items are hard to obtain, if not impossible [63, 61]. The radio edit or live version of given song, for instance, could be mistakenly delivered instead of the expected album version of the same song; a movie could be an unrated version, or a remake of the same story. In fact, even the most precise description of an indescribable item would still leave room for misinterpretation [19, 63, 61]. This fact alone makes trustee-based validation unsuitable for this type of items, and complicates dispute resolution greatly – as well as fair exchange protocol design.

Stating that a univocal description of an item will be available for parties is, therefore, a dangerous assumption usually made by most fair exchange protocol designers. If one or more of the items being exchanged are indescribable, current approaches that rely on previously-obtained/public descriptions are unable to guarantee that item validation will be robust enough for allowing a party to predict the outcome of the transaction. Since indescribable items – particularly digital music and other forms of multimedia content – are currently of great interest to several e-commerce providers – which usually rely on some instantiation of fair exchange protocols – describability arises as an important issue for future research. In fact, indescribable items have only recently been identified as the protagonists of problematic exchanges and, as such, have received some attention [19, 63, 61].

2.5.3 Generatability

Since failure in providing a desirable outcome to all parties in exchanges concerning digital items can be rather difficult to resolve (mostly due to the idempotency property, discussed in Section 2.5.1), most optimistic fair exchange protocols usually rely on some level of

generatability, which can be embedded in items. An item is said *generatable* if a party is able to obtain an equally satisfying item – possibly a copy of the intended item, or a different item which substitutes it in every aspect of interest – with the help of a TTP, provided that the affected party is able to prove her commitment to the transaction. An example of generatable item would be a signed contract by both parties of an agreement, which could have the same legal value if signed instead by both one of the parties and the TTP.

Generatable items have received a lot of attention since the proposal of fair exchange protocols. As stated in [58], an item is said to be *generatable* if it "*can be generated by the trustee in case the receiving party can prove that it has behaved correctly*". The strength of this generatability is defined over the possibility of success: *strong generatability* ensures that the trustee will always be able to generate the item successfully, while *weak generatability* allows failure in the generation, in case of party misbehavior; in such cases, the trustee is able to detect and provide evidence of this misbehavior to the honest party, so that external disputes may be initiated.

Although generatability is not an inherent property of digital items, it can be achieved with the help of several known techniques [82, 16]. In the remainder of this section we revisit a few solutions presented by Vogt et al. [82].

1. *Strong generatability of generic items (with active TTP):* The first approach relies on an active (online) trustee, and can be achieved by the owner party P sending the item i_P, together with description $desc(i_P)$, to the trustee; the trustee then checks the item against the description and, in case of success, stores i_P during the remaining of the exchange. The trustee also signs the provided description and returns this $SIG_{TTP}(desc(i_P))$ to P, who then uses this term as a proof to counterpart Q that i_P can be provided by the trustee if necessary. Although this approach succeeds in providing strong generatability to any describable item, it requires the trustee to keep a copy of every item for every party it is trusted by, which is completely unpractical for large real-world applications.

2. *Strong generatability of digital signatures (offline trustee):* Another approach relies on verifiable escrow [12] primitives, which are designed to provide strong generatability to digital signatures. This does not require an active trustee, but is restricted to signatures and thus is not straightforwardly applicable to other kinds of digital items.

3. *Weak generatability of generic items (offline trustee):* The last approach considered in [82] also does not require an active trustee, and works well with describable

items. The trade-off is that only weak generatability is achievable[2]. In order to accomplish that, the owner P must encrypt the item i_P with the trustee's public key, and sign both this encrypted term and the item description. The obtained term $SIG_P(PU_{TTP}(i_P), desc(i_P))$ is then forwarded to P's counterpart Q, which is able to verify P's signature. In the event of a dispute, Q would provide $SIG_P(PU_{TTP}(i_P), desc(i_P))$ to the trustee, which would first check P's signature. If the check fails, the trustee considers that Q has misbehaved, since he would be able to detect the failure himself; if it succeeds, the trustee tries to decrypt $PU_{TTP}(i_P)$, and to validate the resulting item against the description $desc(i_P)$. If the validation succeeds, the trustee forwards i_P to Q and, if it fails, it considers that P has misbehaved. This method has been applied on several previously-published fair exchange protocol proposals [69, 68].

All of these approaches allow different types of items to be made arbitrarily generatable, but share one common characteristic: they require a well-defined (i.e., univocal) description of the item, which makes them unsuitable for indescribable items. We believe that in order to embed generatability into indescribable items, it is also necessary to address indescribability issues. In that context, reversible degradation techniques [63, 61] seem adequate.

2.5.4 Revocability

An item is said to be *revocable* if it can be invalidated by a trustee, when specific requirements are met. As with generatability, different levels of revocability may be provided. While the trustee will always succeed in making *strongly revocable* items useless for its receiver, she may fail in revoking *weakly revocable* items; in such cases, the trustee is always sure that the receiver got or can still get the item, which can help further dispute resolution.

As with generatability, it is possible to embed revocability into items, specially in particular contexts – such as digital payment applications. In fact, many electronic payment systems provide some level of revocability to electronic cheques [55, 11]. The combination of generatability and revocability can be particularly constructive for fair exchange protocol design, as we shall further discuss in Section 4.2; for instance, a trustee may try to generate a weakly generatable item for a cheated party, and in case of failure, she may revoke the issued cheque in order to restore fairness.

[2] In fact, "*no efficient method (i.e., without TTP interaction) is known to make arbitrary goods strongly generatable*" [82].

2.5.5 Forwardability

The first proposed fair exchange protocols assumed items to be **forwardable**. An item is said to be forwardable if "*P can send the item directly to Q, or it can send it to the TTP; the TTP will be able to verify the correctness of the item with respect to the stated description and either store it or resend it to Q*" [10]. According to this original definition and the arguments presented so far in this section, we conclude that an item is said to be forwardable if it is both *idempotent* and *describable* – which also leads us to conclude that, although simple bit strings with no further meaning, for instance, may be considered forwardable [3] (as stated in Asokan's original work), this might not always be the case for more complex digital items – which is usually the case in real-world transactions.

As with describability, assuming digital items to be forwardable may be dangerous. Several proposed fair exchange protocols for generic items follow the model established by Asokan, and also overlook this issue [88, 29, 45, 47] – ultimately resulting in security flaws that can be explored by an attacker [60].

2.5.6 Co-dependency

In fair exchange protocols, items are usually unrelated in any way – which usually means that their values are not linked to each other. There are, however, particular transactions in which one item is only valuable to the interested party if the other item delivered in the exchange is also valuable to the counterpart. We regard these special items as being **co-dependent** from one another.

Examples of exchanges concerning co-dependent items may be found in contract signing protocols. In such contexts, parties are usually interested in obtaining their own signature, as well as the counterpart's, on some digital contract C. In the case of a two-party contract signing, for instance, P would be interested in obtaining $SIG_P(SIG_Q(C))$, while Q would desire to receive, say, $SIG_Q(SIG_P(C))$.

Since both items depend on the same basic information C to be constructed, a misbehaving party might find it particularly difficult – as opposed to exchanges involving generic items – to tamper with her own item in order to end up with a valid item in exchange for nothing valuable (i.e., garbage-for-gold attacks [60]); most tampering attempts would in general result on both parties receiving invalid items $SIG_P(SIG_Q(C'))$ and $SIG_Q(SIG_P(C'))$, if so – which does not violate fairness requirements. Therefore, fair exchange protocols concerning co-dependent items might be easier to design – at least

[3] If an item is essentially a particular bit string with no particular complex function – as opposed to a multimedia file, for instance, which brings inherent perceptual information in it – a cryptographic hash would be enough to univocally describe it. However, bit strings with no particular function are rather rare in real-world applications for exchanging digital items.

in some level – since these so-called garbage-for-gold attacks would be harder to perform (notice that this might not be the case for other attacks – such as that of a party mischievously abandoning the transaction before sending her own item and after receiving her counterpart's).

Capítulo 3

Reversible degradation

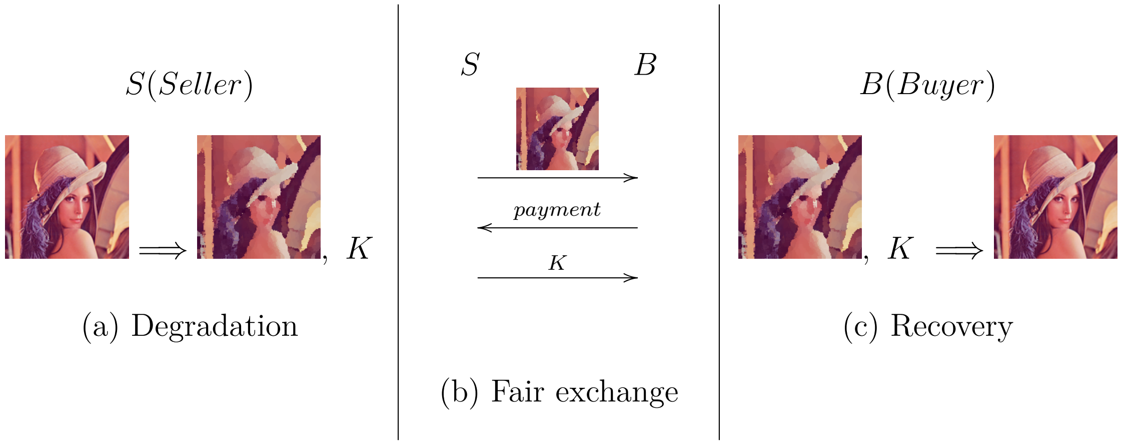
Reversible degradation techniques [63] were proposed as an enhancement to fair exchange protocols with the purpose of enabling them as practical solutions for today's e-commerce needs. By introducing item validation into the critical path of purchase transactions, protocol designers can deploy fair exchange-based e-commerce applications that take into account specific properties of digital items, as opposed to regarding them as generic bit streams. By doing so, it becomes possible to address inherent issues – such as indescribability, for instance – in an integrated fashion.

For the remainder of this section we consider a two-party buying/selling transaction that can be modeled as an exchange of two digital items: an indescribable product of perceptive nature (multimedia content, such as audio, video or picture); and an instance of (possibly revocable) digital payment.

The proposed concept focuses on providing atomic validation artifacts, by taking into account the item's perceptive functionality in order to circumvent their inherent indescribability and achieving some degree of generatability. We revisit the original reversible degradation concept and implement a deployable instance suitable for multimedia content, based on systematic error correction codes (SECCs) [70].

3.1 Concept description

Reversible degradation is conceptualized as follows: by transforming (degrading) the item in such a way that it becomes clearly deteriorated, but can still be distinguished by the receiving party, the owner (which from now on we shall refer to as *Seller*) becomes able to release it for validation without the risk of being cheated. By making this degradation process **reversible**, the counterpart *Buyer* can then receive the degraded copy (an atomic validation artifact, as introduced in Section 2.4), validate it and then negotiate a key for recovering the original, full-quality item. Figure 3.1 illustrates the complete concept.



(a) Degradation

(b) Fair exchange

(c) Recovery

Figura 3.1: Reversible degradation concept description.

The main advantage of this approach over other alternatives is that it allows the buyer to obtain the item before paying for it, and validate the item taking into account its functional (in this case, perceptive) aspect – even if it is somehow degraded. Even though previous e-commerce-oriented fair exchange research has acknowledged before-payment product delivery as a requirement for guaranteeing fairness to the buyer side [14, 54, 15, 6], the currently published approaches often rely on delivering the item in an encrypted form to the buyer (thus treating it as a generic bit stream), and as such, fail to take the product's functional aspects into account – thus undermining item validation and the chances of a successful transaction [65, 63].

Upon receiving the degraded copy, validation can then be performed by *Buyer* relying on his own senses, since human perception is capable of overlooking degradation for the purpose of deciding whether that artifact corresponds, in fact, to the desired item. Also, as we shall discuss in Section 3.5, reversible degradation may, in some cases, require significantly less bandwidth cost than other methods do – and does not rely on costly hardware changes on either end of the transaction to be implemented.

Another important property of the reversible degradation concept is that, after the degradation is reversed, the item is fully restored to its original form – no trace of the degrading information, perceivable or not, is left behind [63]. This differs in essence from DRM methods and some removable watermarking techniques [44] that still leave information embedded into the item and often require the key on every subsequent access to the content. Such techniques require the user to employ special software or hardware to consume the purchased product, which is in general a problem if the user prefers a specific software player or intends to use a particular audio file on his portable digital music device.

Several paradigms have been pointed out [63] as good starting points for implementing reversible degradation – such as error correction codes [50], removable watermarking [44, 39] and perceptual cryptography [42]. In Section 3.2 we detail how SECCs – specifically, Reed-Solomon codes [70] – can be used to implement reversible degradation for a particularly-structured class of multimedia items.

3.2 Implementation of reversible degradation with SECCs

In this section we present an instantiation of reversible degradation [63]. We begin with a brief introduction on error correction codes, focusing on their relevant aspects to our application, and describe the key derivation, degradation and recovery processes in a detailed fashion.

3.2.1 Error correction codes (ECCs)

Error correction codes (ECCs) [50] are regarded as an important research area of Coding Theory, and deal with controlling errors in data transmission over unreliable/noisy channels. This is usually achieved by preprocessing the information to be transmitted in such a way that some degree of redundancy is introduced to the original data; this redundancy allows the receiver to later efficiently detect and recover the information originally intended by the sender – given that the amount of errors introduced by the channel (that is, the number of bits flipped due to channel interference) is smaller than some predefined threshold.

Aside from the obvious advantages of information recovery when errors may occur during data transmission, ECCs have been widely studied as the base for cryptographic algorithms [74, 50, 48]. In general, such algorithms rely on the hardness of the problem of efficiently recovering information from pseudorandomly disturbed data without additional information.

As an example of how ECCs can be used to originate cryptographic artifacts, let us consider the McElice cryptosystem [48] – an asymmetric encryption algorithm which has been widely studied due to its alleged suitability for post-quantum cryptography. This cryptosystem relies on a special class of geometric codes – the binary Goppa codes [31] – and can be summarized as follows: First, *Alice* generates a private key and the corresponding public key. In order to accomplish that, *Alice* first chooses a suitable Goppa code and determines its generator matrix G, capable of correcting t errors. *Alice* then disguises G as a general linear code, by deriving the matrix $G' = SGP$, where S is an invertible matrix and P is an invertible permutation matrix. *Alice* reveals (G', t) as her public key and keeps (S, G, P) as her secret private key. From this point on, if a party *Bob* wants to communicate a sensitive message m to *Alice*, he should encrypt m first by computing a vector $c' = mG'$, and second by generating a random vector z, containing exactly t ones, and calculating the ciphertext $c = c' + z$. The decryption algorithm performed by *Alice* is equally simple: upon receiving c, *Alice* inverts P, thus obtaining P^{-1}, and undoes her own permutation by computing $c'' = cP^{-1}$. *Alice* then uses the error decoding algorithm for the selected code on c'' in order to correct the t errors intentionally introduced by *Bob* during the encryption process, thus obtaining an error-free message m''. Finally, *Alice* is able to recover the original message by calculating $m = m''S^{-1}$.

In the reversible degradation context, one of the originally proposed requirements is that the degraded copy should retain some of the information (and some of the functionality) of the original item in order to empower the receiver to accurately validate the product [63]. For that purpose, a particular class of codes – the **systematic error correction codes (SECCs)** – is more suitable as the basis for reversible degradation techniques. Such codes have the property that a k-symbol long message m will be encoded into an n-symbol long codeword, with $k < n$, in which the first k symbols correspond to the same symbols that compose m; the remaining $(n - k)$ symbols are regarded as *parity symbols*. This means that, for any message $m = (m_1, m_2, m_3, ..., m_k)$, the resulting codeword c has the form $c = (m_1, m_2, m_3, ..., m_k, p_1, p_2, ..., p_{(n-k)})$ – where m_i and p_l are the symbols that comprise m and c for $1 \leq l \leq n - k$ and $1 \leq i \leq k$. As we shall proceed to discuss, this property is crucial for the implementation of our reversible degradation technique.

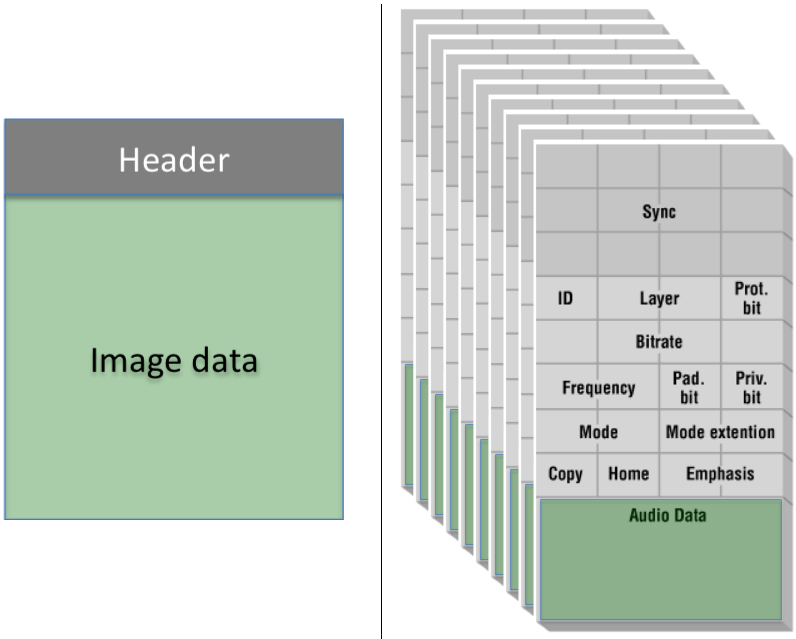
A well-known example of SECC are Reed-Solomon (RS) codes [70]. Such codes are regarded as non-binary and therefore allow symbols to be composed of more than one bit (we shall consider eight-bit/one-byte sized symbols in our explanation[1]). The encoding process regards a k-symbol long message m to be a polynomial $m(x)$ of maximum degree $k-1$, which will be encoded into a n-symbol long codeword c, with $n = 2t + k$, where t is the maximum number of errors (aka, wrong symbols) that can be corrected by the code.

3.2.2 Item characteristics

We assume the item on which the reversible degradation technique will be performed on is of perceptual nature – say, an image or digital audio file – and composed by a sequence of one or more well-identified, mutually independent *frames*. We regard a frame as being well-identified if it is composed by a known-sized *perceptual data block* (e.g., a block of audio bytes or pixels) preceded by a known-sized *header* (which contains information

[1] Even though the proposed technique can be implemented with symbols of different sizes, it should be taken into account that the symbol size affects not only the number of required encoding/decoding operations, but also the efficiency of the technique as a whole – larger symbols mean more expensive algebraic operations.

about that particular frame – such as the size of the succeeding data block). Frames have to be mutually-independent in the sense that a particular header should only contain information regarding its succeeding data block, and no other. Examples of such items are MPEG Audio Layer III (MP3) files [72] and PGM/PPM image files. Figure 3.2 illustrates the required item structure.



(a) PGM/PPM file structure (b) MP3 file structure

Figura 3.2: Examples of frame-like structured items.

3.2.3 Degradation and key generation processes

The following degradation method, which receives the original item as input and outputs a *degraded copy* and a *restoring key* for reversing the degradation, is applied iteratively to each frame composing the item:

1. **Code definition**: First, *Seller* determines the code parameters he will use for the degradation. He chooses the type of systematic code he will use (in our explanation, we assume Reed-Solomon [70] codes); the symbol size s in bits (we suggest $s = 8$ because of the byte model that describes modern computation); the number k of s-bit sized symbols that compose each message to be encoded; and the maximum number t of errors that will be introduced to degrade each message and that the chosen code will be able to correct. With this information, *Seller* can then choose

an appropriate code for this transaction – an $RS(n, k)$ code, where $n = 2t + k$ gives the size of each codeword. He determines an irreducible polynomial of degree s that describes a finite field $GF(p^s)$, with coefficients in $GF(p)$ (p being a prime number; in our explanation, we assume $p = 2$), and a generator polynomial $g(x)$ for the code with s-bit sized coefficients and degree $n - k$.

2. **Header preservation**: The next step for *Seller* is to separate the header from the data block; this will be essential for preserving the functionality of the degraded item, since the header contains crucial information for the interpretation of the succeeding data block (in other words, what kind of information – visual, audio or otherwise – the data block contains).

3. **Data block splitting**: After the header is safely removed, *Seller* proceeds to split the data block into a sequence of k-symbol long *messages*. Each of these messages will be individually encoded into an n-symbol long codeword with the chosen $RS(n, k)$ code on the next step.

4. **Messages encoding**: For each message m_i obtained from the data block, *Seller* produces an n-symbol long codeword c_i with the previously defined code $RS(n, k)$.

5. **Controlled error embedding**: Each codeword c_i is then disturbed in t of its symbols, resulting in a *disturbed vector* c'_i. The disturbed symbols receive a new s-bit value generated randomly by *Seller*.

6. **Disturbed vectors splitting**: Now *Seller* proceeds to split each disturbed vector c'_i in two vectors: a *degraded data vector* d_i, composed of the first k symbols of c'_i, and a *parity check vector* v_i, composed of the remaining $n - k = 2t$ symbols of c'_i. Notice that, because the code is systematic, each non-disturbed symbol in d_i is identical to its corresponding in m_i.

7. **Degraded item assembly**: *Seller* then concatenates each degraded vector d_i, in the same order the corresponding m_i's appeared in the original data block – thus composing a *degraded data block*, which is identical to the original data block except for the disturbed symbols. He also concatenates the degraded data block to the original header. We refer to the output for this step as a $[(t/k) * 100]$*%-degraded version* of the original item.

8. **Restoring key assembly**: Finally, *Seller* concatenates the defining parameters for the $RS(n, k)$ code (given by s, t, k, p, $GF(p^s)$, $g(x)$) and each of the parity check vectors v_i. The output for this step is regarded as the *restoring key* (or simply *key*,

from now on) required for reversing the degradation process we just described, and is kept secret by *Seller* until payment has been completed.

3.2.4 Reversion process

After engaging *Seller* in the purchase protocol, and provided that the validation performed on the degraded version of the item was satisfactory, *Buyer* will pay in exchange of the secret key for restoring the item to its original quality. By using an error-decoding-capable piece of software provided by *Seller* – which takes the degraded version and the key as input – *Buyer* is able to generate an error-free, clean version of the item from the degraded copy, which will be identical to the original product used by *Seller* as input to the degradation process: he restores each disturbed codeword from the degraded data block and the parity check vectors in the key and, by using the provided information about the code, is able to correct the errors and to compose a *restored data block*. This restored data block can be finally concatenated to the header in order to obtain a *restored version* of the item.

We note that, by including the code parameters in the restoring key as part of the restoring key, our method allows *Seller* to make transaction/product-dependent decisions regarding the degradation level he desires to apply. We proceed to discuss how transactions are affected by *Seller*'s choices in Section 3.3.

3.3 Considerations about the technique

In this section, we evaluate our proposal regarding some relevant aspects.

3.3.1 Flexibility: controlled degradation level and nature of content

One important feature of our technique is that the degradation level can be arbitrarily controlled by *Seller*. The amount t of errors to be introduced to each message impacts significantly not only the quality of the degraded version, but also the security (more errors are harder to correct without the key than fewer errors) and both computational and transmission costs (the more errors are introduced, the more expensive the encoding/decoding operations will be and the larger the key will be). The choice of a suitable value for t should take all these aspects into account and depends on both the perceptual nature and individual characteristics of the item to be degraded (as we shall further discuss in Section 3.4).

This allows *Seller* to take commercial needs and quality of service aspects into account – for instance, a cheap file can be less degraded in order to improve delivery efficiency, since the amount of resources required for "attacking" the degraded version will be too large in comparison to the item's price; a more-expensive item, on the other hand, can be more strongly degraded in order to ensure that it will be virtually impossible for an attacker to restore its full quality without paying for the key; still, the opposite approach could also be valid – *Seller* could choose to offer better quality (i.e., less degraded) versions for validation of expensive items in order to provide *Buyer* with stronger guarantees of a satisfactory acquisition – thus aiming at attracting new consumers into higher-priced transactions. In summary, the proposed method suits several buyer/seller relationship models, and do not interfere, in that context, with *Seller*'s pre-established business decisions.

Another advantage of our technique is that it can be applied regardless of the perceptual nature of the item, and can be equally implemented for transactions involving (un)compressed audio, image or movie content – with the restriction that the item must be presented in a frame-based file structure, as described in Section 3.2.2. This is of particular interest to multiple-content providers interested in offering several kinds of media on their product catalogs, since it avoids the need of implementing distinct degradation processes for each kind of media.

Finally, the technique allows the full recovery of the item, provided that the restoring key is known, as suggested by the original reversible degradation guidelines [63]. Because the restored item will be identical to the original one, the degradation process can be performed on top of other general purpose techniques – such as compression, watermarking, DRM, etc. This is not only relevant for the *Buyer* – who desires to recover the item to the best possible quality – but also convenient for the *Seller* interested in the benefits introduced by such technologies.

3.3.2 Efficiency: relationship between key size and degradation level

Concerning efficiency, since all the described methods only include basic algebraic operations over a defined finite field, it becomes obvious that the complexity of all the described methods is polynomial in k, n and t. However, as it is often the case in cryptographic applications of error correction codes, the restoring key size can be relatively large in comparison to item size. An item composed of f frames, each composed of u s-sized symbols (s is given in bits), will result in a key size given by $[f(u/k)(n-k)s+w] = [f(u/k)(2t)s]$ bits, where w is the number of bits taken by the code parameters included in the key; since the item size is given by $f(u+h)s$, where h is the frame header size, the approximate ratio between key and item size – considering w and h to be negligible – is given by Equation 3.1.

$$\frac{\frac{[f(u/k)2t+w]s}{k}}{f(u+h)s} = \frac{2tu+w}{k(u+h)} = \frac{2t}{k} \tag{3.1}$$

Equation 3.1 shows that the relationship between item and restoring key sizes is controlled by the intended degradation level (i.e., the amount t of introduced errors per message, which obviously also affects the size k of each message to be encoded). This leads to the fact that, even though larger restoring keys might be expected when larger items are degraded, this is not always the case – since the perceptive nature of the item also plays a role on the required degradation level for effective *Seller* protection. We further discuss how the restoring key relative size may vary for different types of content (and different degradation levels) under the light of our experimental data, in Section 3.4.

Nevertheless, specifically when lower degradations levels are used, the restoring key can be significantly small in comparison to the original item size. In such cases, and particularly if large items are being sold/purchased (such as digital movies), this would represent significant bandwidth saving in the protocol – comparatively to the use of non-atomic validation artifacts, which would require a large amount of data to be downloaded more than once by *Buyer*. Even though this is a circumstantial and instance-dependent advantage, we believe it to be still worthy of notice and relevant to specific real-life application scenarios.

3.3.3 Seller-side security: robustness against brute force attacks

Another important aspect that we must take into account is security. Much like the encryption/decryption correlation in traditional cryptosystems, removing the introduced degradation should be easy to perform (provided that the key is known) but otherwise hard to accomplish [50]. In the proposed technique, all information about the code (including the parity check symbols) is kept secret by *Seller* until the payment is performed. In other words, in order to be able to undo the degradation process and restore the full quality of the item without the information that composes the key, a malicious *Buyer* would have to be able to detect and correct the wrong symbols on each frame of the received information – a task known to be hard in the general case of error correction codes [37]. We proceed to discuss how effective brute force attacks would be in the specific context of our application.

Let us suppose that a malicious *Buyer* intends to perform a brute force attack on the degraded version of an item, with the goal of removing all (or at least a significant part) of the introduced errors in order to obtain a suitable (free) copy for consumption. For this purpose, let us consider the item to be an MP3 file (its file structure is presented in Figure 3.2b).

The only information available to *Buyer* at the moment of the attack comes from the degraded copy itself. His first step would be to split the file into a collection of individual frames, which he would then proceed to analyze individually; this is true because, since the frames are mutually-independent by specification [72] and each error was introduced as a random value in a random position of each frame's data block (and as such, are not correlated in any fashion), completely restoring one particular frame does not provide the attacker with any information on how to restore any of the remaining frames. For the remainder of our explanation we assume the item to be d bytes in size (for MP3 files, d is in the order of megabytes), distributed into f frames. Therefore, each frame is composed of $d_b + h = d/f$ bytes, from which the first h bytes comprise the header and d_b bytes comprise the data block – containing the actual perceptual (in this case, audio) information of that particular frame.

Now, let us suppose that f_m is the number of messages in which the data block of a particular frame was split during the degradation process (Step 3, Section 3.2.3). Each of these $f_m = d_b/k$ messages is composed by k symbols (bytes, in our implementation), and is disturbed by t randomly-chosen/placed errors. Therefore, the number of errors required to be detected and corrected by the attacker, in order to restore one full frame from the degraded copy, is (taking into account $h \ll d_b$ – MP3 frame header size is fixed as 4 bytes, for instance [72]):

$$t' = tf_m = t\frac{d_b}{k} = t\left(\frac{\frac{d}{f} - h}{k}\right) = \frac{td}{fk} \tag{3.2}$$

In practical terms, let us suppose that the attack is to be performed in one of the provided MP3 audio files we used during our tests – namely the song *Help!*, by *The Beatles* (filename *help_andnow.mp3* in the provided experimental data [1]). Relevant file characteristics are: 320 Kbps (40 KB/s) constant bit rate; file size $d = 5738579$ bytes; duration 143 seconds; $f = 5492$ frames. The copy obtained by the attacker is degraded by a factor of 0.2% (filename *help_andnow_deg_0.2percent.mp3* in the provided experimental data [1])), which means that $d_b \approx 1040$ bytes. With these parameters, the number of introduced errors per message is $t = 2$, giving us messages of $k = 251$ bytes for generating codewords with $n = 255$ bytes. Therefore, each frame's data block is encoded as $f_m = 4$ messages, each of them being disturbed by $t = 2$ errors – in a total of $t' = 8$ errors per frame.

Proceeding with our hypothetical brute force attack scenario, restoring one single frame would mean to be able to find which eight of the 1040 bytes in the data block were disturbed, and which were the correct values before degradation. Even if we suppose the calculation of all restored candidates for one particular frame to be a feasible task, the attacker would still have to be able to identify which of the restored candidates is the correct one. In other words, such frame-by-frame brute force attack would require from the attacker the ability to perceive, upon successfully guessing the eight original values of the disturbed bytes in a frame, some noticeable quality gain on the degraded copy – so that he could learn when to move on to the next frame and repeat the process until all frames were restored. However, each frame accounts (with these parameters) for approximately 0,02 seconds of audio playback. We firmly believe that noticing any quality gain in such a small time window is beyond the capabilities of even the most well-trained audiophile.

We also notice that, if instead of a frame-by-frame brute force approach, the attacker had chosen a byte-by-byte (that is, error-by-error or symbol-by-symbol) approach, the same argument applies: with a total of $d_b * f \approx 5711680$ bytes of actual perceptive data (that is, headers excluded), each possibly-disturbed byte accounts for no more than $0,000025$ seconds of audio playback. A general estimation of the duration (in seconds) of the portion of audio affected by one single error in MP3 items is given by Equation 3.3.

$$D_{error} = \frac{1}{bitrate} \tag{3.3}$$

As a side note on our view on the feasibility of brute force attacks regarding the proposed method, we point out that, even though we have chosen to take an instance-based approach to the subject and, as such, provide instance-dependent values to illustrate our claims, Equations 3.2 and 3.3 support our conclusions even in the general case. Equation 3.3, in particular, shows that brute force attacks are not feasible with real-world MP3 files (which are commonly commercialized with 128 Kbps (16 KB/s), 256 Kbps (32 KB/s), 320 Kbps (40 KB/s) compression bit rates, for instance) – even if a relatively small degradation level is applied to the item. In summary, this happens due to the fact that each “hit” accomplished by iterative brute force guessing will represent a too-slight quality gain in the attacked degraded copy – so slight in fact that the (malicious) listener will be unable to perceive any progressive success of his attack iterations.

Finally, let us briefly consider how the hereby presented discussion on brute force effectiveness would be affected if the item of interest was a single-frame PGM/PPM image file – instead of a multiple-frame MP3 audio file. Not only PGM/PPM files differ from MP3 files in the amount of frames in which the perceptive information is structured, but also in how that information is perceived itself. In other words, while an audio file is “consumed” by the listener throughout a period of several seconds – meaning that only a “portion” of the whole content is perceivable by the hypothetical attacker in any particular moment – this is not true for image files, which are perceived by the viewer as a whole and instantly. Therefore, it should be arguably easier for the attacker not only to identify the location of a single introduced error (since each symbol is, in this case, a pixel) – by spotting pixels that differ too much from their neighbors – but also to perceive some slight quality gain in each brute force guessing iteration.

We believe this to be the reason why image items seem to require higher degradation levels in comparison to MP3 files (as we shall more thoroughly discuss in Section 3.4). Furthermore, while in the latter scenario degradation is applied on top of compression (thus causing each introduced error to be expanded during audio playback and to generate more perceivable noise), our current implementation with image files degrade the item of interest at pixel level (which means that each error only affects one particular pixel – as opposed to the expansion effect observed in audio content). In summary, we conclude that, by using higher degradation levels on image files (and, as such, introducing more errors relatively to MP3 items), we are able to limit how easier it is for the attacker to recover individual pixels through neighbor analysis.

3.3.4 Buyer-side security: fair exchange and dispute resolution

From *Buyer*'s perspective, security issues take a different form. His concerns are more intimately related to the fair exchange aspects of the transaction – specifically, whether the restoring key he is about to pay for will indeed legitimately reverse the degraded version of the item into a satisfactory version of the same content (as opposed to somehow "transforming" the item into something completely different, leaving degradation traces behind or changing intrinsic characteristics such as image resolution or audio compression bit rate, for instance).

In order to fully understand the advantages of reversible degradation from *Buyer*'s perspective, we must go back to the issues introduced in Chapter 2 and Section 2.4. As we have discussed, the current sale/purchase model for digital items does not enable *Buyer* with a sufficiently unambiguous item validation process, which allows unfair outcomes to happen on some transactions. For that reason, we approach the issue by providing a method for item validation that 1) relies on an atomic validation artifact that partially preserves relevant functional aspects of the final product; 2) takes into account even minimal variations between different versions of the same content; 3) takes into account different expectations *Buyer* may have regarding the characteristics of the final product; and 4) allows dispute resolution to be conducted unambiguously, in cases of exceptions.

Particularly in the context of dispute resolution, an unsatisfied *Buyer* would be required by the TTP to provide the validation artifact, the restoring key and proof of payment, so that the TTP can rule over the validity of his claims. The TTP would then be able to obtain the restored version of the product, which she would then compare with the degraded version taking *Buyer*'s claims into account.

If the claimed differences between degraded and restored versions can be observed by the TTP – which in audio content, for instance, could range from minor lyrics variations to a completely distinct song – she rules in favor of *Buyer*; notice that, for more specific claims, the TTP might require the assistance of an expert on the work of that particular content author, if necessary, as a means to fairly decide the matter. If, on the other hand, the claims cannot be observed by the TTP when comparing both versions, she rules against *Buyer*. In such cases, this decision is supported by the fact that either the validation artifact did not contain enough information for *Buyer* to validate the transaction – in which case he should not have paid for the restoring key but, rather, should have contacted *Seller* for a less degraded version; or, more simply, both versions do not present the claimed differences at all – being verifiably identical regarding those claims.

By providing the base for a less ambiguous dispute resolution process (in comparison to the current model discussed in Chapter 2), the proposed validation method guarantees *weak fairness* – which refers to the power of decision a TTP has to unambiguously decide which party is to blame in the event of an unsatisfying transaction outcome. It has been noted that *strong fairness* – the guarantee that the protocol does not allow for unsatisfying transaction outcomes for either party – is hard and, in most cases, even impossible to achieve [30, 57]. Therefore, by enabling weak fairness through unambiguous dispute resolution guarantees, we believe the proposed method to be – as far as *Buyer* security goes – more reliable than the current model for sale/purchase of digital (indescribable) content.

3.4 Experimental data

In order to evaluate the proposed technique, we implemented a proof-of-concept of the methods described in Section 3.2. We provide the implemented source code and a subset of our test data for further evaluation [1]. In this section, we describe relevant characteristics of the provided data, and discuss our results under experimental analysis.

3.4.1 Implementation details

We implemented our proof-of-concept using the Python programming language (version 2.7.1), extended with a third-party module [34] for Reed-Solomon codes generation/processing (version 0.1). Our implementation accepts either clean MP3/PGM/PPM files as input for degradation, or degraded versions of those files, together with the corresponding, previously generated restoring key file, as input for recovery. In order to keep our implementation as simple as possible, input files have to be stripped of any irrelevant information for the degradation/recovery processes (such as comments or, in the case of MP3 files, ID3-tags containing meta-data).

The choice of Reed-Solomon codes is also fixed for some parameters: The implementation relies on codes capable of processing messages composed by 8-bit (one byte) symbols into 255-byte long codewords (due to restrictions of the third-party module). The amount of correctable errors t (and therefore the degradation level) can be set by the user, but is bounded by the chosen code and its implementation in the third-party module: $t \geq 0$, $t = (255 - k)/2$ and $k \leq 128$ (all values given in bytes). Therefore, for this simplified proof-of-concept, the maximum amount of errors that can be processed per message is $t = 64$, which produces a 50%-degraded output.

3.4.2 Package contents and items description

Two sets of items are available for evaluation: a set of *Image data* and a set of *Audio data*. Each set includes examples of different (but similar) instances of the corresponding content type; both clean files without any degradation (*Original files*) and degraded versions coupled with the corresponding restoring keys (*Degraded versions + keys*) are provided. For completion, we also included restored versions generated from the reversed degradation process when each provided degraded version is used as input with the corresponding provided restoring key (files with suffix “*_rest*” in the *Degraded versions + keys* folder). Finally, the package includes the source code of our proof-of-concept implementation (*degradation.py*) and the batch script used for generating all the provided examples (*testgen.bat*).

To illustrate PGM/PPM degradation effects, we chose three instances of pictures of model Lena Söderberg: the well-known Lena’s portrait, commonly used as an example in Visual Processing research (*lena_color1.ppm*); a gray-scaled version of the same portrait (*lena_gray.pgm*); and a different, less-known picture taken from the same photo session (*lena_color2.ppm*). These three versions were chosen under the assumption that – even though they are indeed different representations of Lena, and as such would constitute different products in a hypothetical *Seller*’s catalog – the three items hold enough similarity and context-sensitive relationship, which could lead to inaccurate item validation on *Buyer*’s behalf (as previously discussed in Chapter 2).

As for MP3 evaluation, we chose a similar approach: the song *Help!*, by *The Beatles*, is both widely famous and available in several different versions (or, in this case, *takes*). Three of the four provided takes were recorded in studio between April and May of 1965 and released in 1999 as *The Alternate Help!* bootleg, a collection of demo, live and alternate versions of several of *The Beatles*’ songs; the fourth take was recorded for the *Help!* movie release. They differ from each other as follows:

- **Take 5 (*help_inst.mp3*)**: This take includes no vocals, being purely instrumental instead.
- **Take 9 (*help_double.mp3*)**: This was the first recorded take featuring vocals. The knowledgeable *The Beatles* fan will recognize it mainly by two aspects: 1) it features doubled backing vocals (as it can already be noticed in the first two seconds of playback), as opposed to the single backing vocal version originally released in 1965's *Help!* album; and 2) it features a tambourine, overdubbed in the same track as the vocals, among the percussion instruments.
- **Take 12 (*help_main.mp3*)**: A very similar take to the one included in *Help!*'s original album release, with the exception of a slight tempo variation (noticeable in the vocals, which sound a little lower-toned than in the final recording). Also, in this version both a rather subtle guitar chord and the words "*one, two, three, four*" can be heard before the song begins.
- **Movie take (*help_andnow.mp3*)**: The vocal track for *Help!* had to be re-recorded for the opening sequence of the 1965 movie with the same title; even though the reasons for that are still argued by both fans and experts, the commonly-accepted explanation is that, while the available master tape included the overdubbed tambourine/vocal track, the movie's opening sequence featured the band playing the song without a tambourine. For that reason, the movie version of *Help!* includes subtle unintentional variations in John Lennon's main vocals that, while passing unnoticed to the average listener, are well-known and evident to the hypothetical *connoisseur* engaging a purchase transaction as *Buyer*: For instance, at 00:21 of playback time, the words "***and** now these days have gone*" are uttered, instead of the "***but** now these days have gone*" present in other takes. Another example happens at 00:26 of playback time, when the spacing between words in the verse "*now I find I've changed my mind*" is also different from the one in other known takes.

As a final note, we have chosen to illustrate our degradation technique with arguably subtle different versions of the presented contents – instead of more obviously-different versions. In other words, even though the variations among the versions included for comparison may be deemed irrelevant for most consumers, we consider the discussion of what are (ir)relevant differences among similar items to be an abstract/personal one and, as such, aim at providing a method that is able to guarantee item validation for transactions involving even the most seemingly indistinguishable product instances and most demanding consumers. We also note that this same argument – with the addition that most hereby presented variations (specially in the *Audio files* set) are not even known to exist by the average consumer – is the main reason behind the issue presented in

Chapter 2 and illustrated by the consumer complaints in Figure 2.5: because product versions in this context can be as subtly different as they can be varied, and assuming that in most cases the textual product description in *Seller*'s catalog will not be written by a *connoisseur*, the current business model is prone to human misinformation that can lead to varying degrees of consumer unsatisfaction.

3.4.3 Degraded versions

We also provide, for each included degradation-free (clean) version, a few pairs of degraded copy/restoring key produced with variable degradation levels. For MP3 files, the included degraded versions were produced using 0.2%, 2% and 10% degradation levels – while PPM/PGM files were degraded with 25% and 50% levels; as we introduced in Section 3.3.3, we believe that single-framed image content requires (and at the same time allows validation under) larger degradation levels due to their "consumption-as-a-whole" perceptive nature – as opposed to multiple-framed compressed audio content, which is consumed throughout several seconds of playback.

In order to illustrate this claim, we observe in our tests that 10%-degraded versions of audio content are barely identifiable as a song at all – regardless of compression bitrate or song length. Image content, on the other hand, is still perfectly recognizable in 25%-degraded versions and, more arguably, even when 50% of the visual information is disturbed. Regarding lower degradation levels, we observed that 2%-degraded versions of our audio tests still retain enough information to allow more obvious version differences to be recognized (such as the lack of vocals in *Take 5* and the tempo variation in *Take 12*), but it still introduces too much noise to allow for spotting more subtle variations (such as the tambourine featuring in *Take 9* and the lyrics change in the *Movie take*). Under those observations, and taking into account the discussion on security presented in Section 3.3.3, we conclude that degradation levels between 0.2% and 1% should be enough to provide both validation capabilities to $User$ and robustness against brute force recovery to *Seller* concerning MP3 content. In the same fashion, degradation levels between 25% and 50% seem to be suitable options for PGM/PPM content (Figure 3.3 illustrates the included clean and degraded versions for the three included versions of the portrait of model Lena Söderberg).

Regarding restoring key sizes, we observe that the discussion conducted in Section 3.3.2 applies. Therefore, as shown by Equation 3.1, 25%-versions are approximately half in size as the original item, reaching the same size at 50% degradation level. For audio content, on the other hand, key sizes will be considerably smaller than the product size – since lower degradation levels are required/allowed for this type of content.

However, even in scenarios when high degradation levels are required – thus nullifying

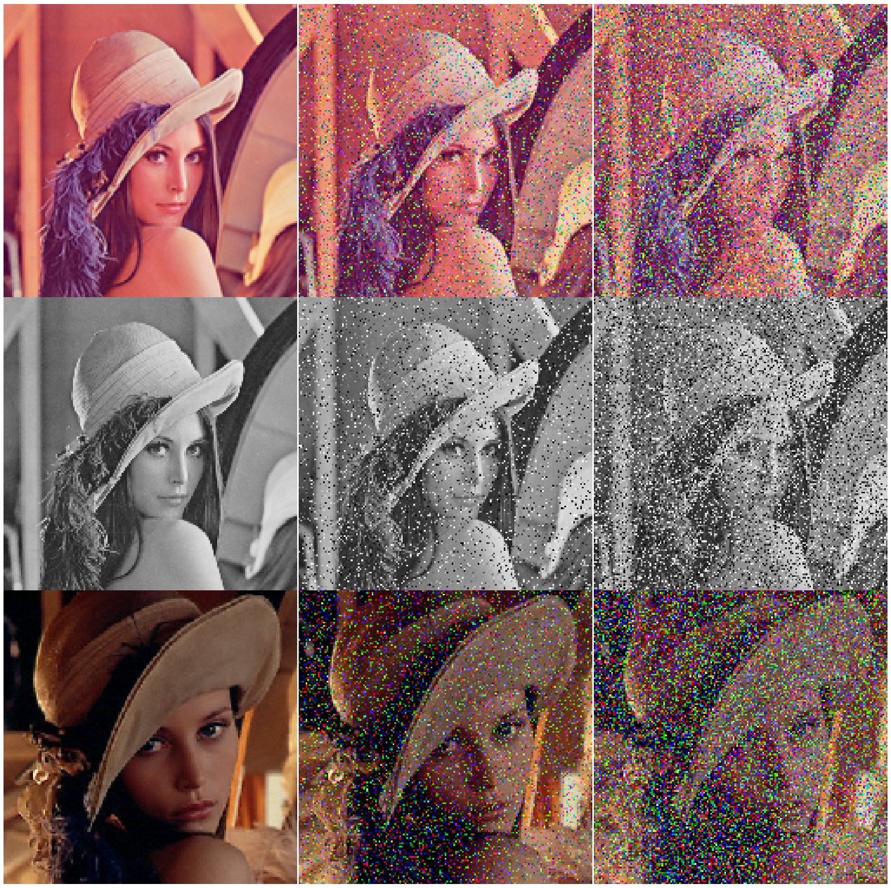


Figura 3.3: Three different 169x169-pixels pictures of model Lena Söderberg, and their corresponding 25%-degraded and 50%-degraded versions. The first and the third files are in PPM format, while the second one is in PGM format.

any bandwidth savings that could otherwise be accomplished with the proposed technique – reversible degradation stands as a suitable alternative model for online buying/selling digital content, providing other less arguable benefits in comparison with other approaches. We shall further extend this argument in a comparative discussion regarding alternative models, in Section 3.5.

3.5 Alternatives to reversible degradation

In this section we present a few alternative models for the purchase/sale of multimedia items. We also point the main problems with each of them, and compare them to the reversible degradation model.

3.5.1 Sample preview with embedded player

This is the solution currently adopted by many digital content providers to enhance item validation of indescribable products, as described in Section 4.3. Unfortunately, as previously discussed and illustrated by the examples in Figure 2.5, offering a (usually limited) preview sample together with the list-based features description of the product is often not enough to provide the buyer with the required knowledge to accurately validate the item, thus leaving room for mistaken product purchases.

One could point out that, in order to allow for precise comparison between several similar versions of a product, full preview of the content might then be made available by the store before payment – thus reducing the problems that might arise if a bad section is chosen as a preview in the limited sample preview model. By listening to the whole song or watching the whole movie before buying, customers can better decide if that item is really what they are looking for. However, this would allow any malicious user to easily capture the video or audio stream and record it without any quality loss, thus obtaining the item without having paid for it (due to the idempotency property of digital items, discussed in Section 2.5.1). This would be unfair by definition, and might become the cause of financial loss for media sellers.

3.5.2 Random sample preview with embedded player

If instead of providing a limited, fixed preview of the item, sellers were to provide a randomly, dynamically-chosen portion of the file to be previewed every time the preview button was clicked, the bad portion problem of limited preview would also be solved. And since the full item would not be provided to the user, recording the stream to a new file would be harder.

Harder, but not impossible. Even if only a limited, randomly chosen portion of the item was made available each time the customer clicked on the preview button, a full copy of the file would not be that hard to obtain. By clicking the preview button several times, and by recording each sample of the product, it wouldn't take long before the malicious user could obtain the whole item separated in several recordings. Reconstruction of the whole item would then be easily accomplished with a simple software editor.

We should remark that even if this system were smart enough to never select a particular portion of the item to be previewed – thus avoiding full item reconstruction by trivial methods – there could be cases in which the hidden portion was exactly what the user wanted to preview in order to decide whether the item is the one he desires to buy. Some songs, for instance, have too many versions that can be very similar to each other, differing only by a few seconds from one another. The same might happen with a director's cut release of a movie, that may come with a few additional frames than the regular version. In such cases, hiding a particular portion of the item might still present the buyer with an undecidable situation.

3.5.3 Lower quality samples

Instead of providing a preview of any kind, stores could provide the buyers with a full, lesser-quality version of the item for download before payment. This approach is very similar to reversible degradation, except that the lesser quality (aka, degraded) version is not reversible and is fixed as a deliberately lower bit rate validation artifact. Regarding image files, the equivalent would be a low resolution version of the item [19].

This approach does not, however, rely on atomic validation artifacts. As introduced in Section 2.4 and further argued in Section 3.3.4, when items of complex nature are concerned (as is the case with products of perceptual nature, which are indescribable due to the difficulty in predicting the various user's expectations regarding the characteristics of what he is about to purchase), fairness can only be achieved by means of providing an unambiguous dispute resolution process. By relying on a separate validation artifact for validating the final product, the current model fails to allow an external judge to accurately decide, in unsatisfying transactions, which party is to blame. For instance, the validation artifact and final product provided to the TTP by *Buyer* for comparison could arguably come from distinct transactions – if a dishonest *Buyer* were to try to cheat the TTP into ruling against an honest *Seller*, possibly with the purpose of trying to obtain two distinct products while having only paid for one. On the other hand, regarding our method, if the restoring key and degraded version provided by a dishonest *Buyer* came from different transactions, the TTP would not be able to restore the clean version of the item. She would then contact *Seller*, who would be able to provide the correct restoring key to the TTP – who would then be able to proceed as described in Section 3.3.4 and decide the matter without requiring *Seller* to deliver a second item to *Buyer* (forwarding the correct key to *Buyer* would also be an option in this case).

We also note that this approach would require two times more bandwidth for each purchase transaction in every case. *Buyer* would have to download each item twice – first, the low quality version; and finally, the full quality one – instead of just one item download and a (possibly smaller) restoring key, as it is the case with reversible degradation. Specifically in the case where large files are offered for purchase, such as digital movie content, this would lead to significantly more inefficient transactions – specially when lower degradation levels are already secure enough to provide *Seller* with guarantees against brute force reconstruction.

Finally, we consider the discussion on how worthless a free lower quality sample might actually be a very subjective one. Specifically, while the degradation introduced in the validation artifact by the proposed method does in fact effectively corrupt the item perceptively – even when low degradation levels are used – thus minimizing the odds of a hypothetically non-demanding *Buyer* simply keeping the degraded copy for future consumption for free, this would certainly not be the case if lower quality samples were used instead, as validation artifacts. In fact, many users do not mind relying on cheaper devices (such as digital audio players, headphones, speakers, etc.) for consuming digital content, which do not account for, by design, any noticeable quality gain regarding lower or higher compression bit rates. We believe this is exactly the reason why this method is currently not widely implemented – since even the lowest quality sample (regarding bit rate compression, in the context of compressed audio and video) would still be good-enough to be kept by a significant parcel of consumers, thus discouraging payment for a better-quality version. In summary, the degradation provided by merely reducing compression rates (or image resolution, when image is concerned) is simply not perceivable enough to provide validation artifacts deemed worthless by most consumers.

Therefore, reversible degradation not only enables the transaction with atomic item validation, by allowing the final product to be recovered from the validation artifact itself – thus achieving at least weak fairness by establishing the grounds for unambiguous dispute processes – but can also be arguably more efficient, from the bandwidth consumption perspective, in cases in which lower degradation levels are required.

3.5.4 DRM-based expiration date

Instead of degrading the file, the store could provide the user with a full quality download before the payment was made. The provided item would be rigged with some form of expiration date mechanism that would be triggered after some time, unless otherwise disabled by the user (with some key provided after payment). In other words, digital items could be made revocable, as described in Section 2.5.4, much like it happens with time-based trial periods of demonstration software.

The problem here would be similar to the one discussed in Section 3.5.1. By playing the rigged item before the expiration date, the malicious buyer could simply record the output to another file without any perceivable quality loss. Besides that, DRM acceptance has been widely debated among digital media consumers, mostly because of interoperability issues [81].

Capítulo 4

Item-aware protocol design

In this chapter we present our so-called item-aware approach to fair exchange protocol design. These results have been previously published in [61].

4.1 Introduction

Even though fair exchange protocols have been widely studied, most designs still follow the same approach as Asokan's original work [10], which considered the exchanged items to be generic bit streams with few or no particular properties of relevance to protocol design. We believe, however, that in most current contexts items do have inherent complexity that may interfere with transactions, and exhibit characteristics that either make the exchange easier, or become obstacles for enforcing successful (fair) outcomes. Those properties (see Section 2.5 for a survey on the most commonly found properties in literature) are usually left aside during protocol design for the sake of "simplicity"of explanation, which often results in proposals that are inaccurate, inefficient or inadequate [49, 87, 89, 59] for most real-world applications – where items are far from generic bit streams. To the extent of our knowledge, only few researchers have taken into account how the nature of the exchanged items may impact transaction outcomes [82, 19, 60, 63, 61].

In this chapter we discuss how complex items with certain properties may be harder or easier to exchange for similarly-complex items. We focus on the interaction between the properties surveyed in Section 2.5 and present a discussion on their possible effects on the outcomes of the protocols designed to exchange them. It is our goal to show that, by focusing on the inherent aspects for the items being exchanged – an approach to which we further refer to as *item-aware protocol design*, as opposed to the conventional *generic item protocol design* – the designer may be able to tackle context-specific problems and to avoid common protocol design pitfalls. We are aware that studies providing formal frameworks for complex tasks such as protocol design may be able to provide more-reliable foundations for further development of the state-of-the-art; regardlessly, several previous authors have been able to contribute to a better understanding of protocol design by providing useful guidelines with similarly-informal discussions on the topic [43, 3, 85, 60], as is the case of this specific contribution.

4.2 Notes on the interaction between properties and impacts for fair exchange

In Section 2.5 we discussed a few of the most relevant item properties concerning several fair exchange-related scenarios. In this section, we discuss how two-party protocol design might benefit from the interaction between two items bearing each of those properties, and try to shed some light over what could be gained or lost from the interaction between them by approaching protocol design in item-aware fashion.

For the remainder of this section, we assume that two parties P and Q wish to exchange two items i_P and i_Q. We also assume that P commits to the transaction first, giving up i_P (to which we further refer as first item) before Q gives up i_Q (to which we further refer as second item). We keep our presentation brief in order to ease further reference, basing our statements on the arguments presented so far.

i_q i_p		Idempotent	Indescribable	Generatable		Revocable		Forwardable	Codependent
				Weak	Strong	Weak	Strong		
Idempotent		1	1, 2	1, 3(a)	1, 3(b)	1	1	1, 4	▬
Indescribable		1, 2	2	2, 3(a)	2, 3(b)	2	2	1, 2, 4	▬
Generatable	Weak	1	2	3(a)	3(b)	7	7	1, 4	▬
	Strong	1	2	3(a)	3(b)	7	7	1, 4	▬
Revocable	Weak	1, 5(a)	2, 5(a)	3(a), 5(a)	3(b), 5(a)	5(a)	5(a)	1, 5(a), 4	▬
	Strong	1, 5(b)	2, 5(b)	3(a), 5(b)	3(b), 5(b)	5(b)	5(b)	1, 5(b), 4	▬
Forwardable		1	1, 2	1, 3(a)	1,3(b)	1	1	1, 4	▬
Codependent		▬	▬	▬	▬	▬	▬	▬	6

Tabela 4.1: Interactions between item properties in optimistic fair exchange protocols (see below for details)

Table 4.1 shows the impact that the properties discussed in Section 2.5 may have on an optimistic two-party fair exchange protocol. Co-dependent items, specifically, are special items that only make sense when considered in pairs – which is why we omitted their comparison with other properties. The values included in each cell of Table 4.1 refer to the following statements:

1. No return policies apply for idempotent items (first or second), so fair exchange protocols should be robust enough to minimize unexpected outcomes. Dispute resolution should be carefully designed.

2. Item validation is hard to achieve for indescribable items. Since strong fairness might be hard to guarantee for both the owner (if this is a first item) and the receiver (if this is a second item), as Asokan's protocols are not inherently equipped for these kinds of items, special-purpose techniques may be required as enhancements for practical deployment [63, 61]. In particular, the item validation step should receive special attention during protocol design.

3. When the second item is generatable, the first party can always be assured that, in case of exceptions, the TTP might be able to help her with obtaining the desired item. Therefore, generatable items are suitable as second items in fair exchange protocols.

   (a) If the item is only weakly generatable, the TTP might fail in retrieving it for the interested party. Therefore, only weak fairness is guaranteed.

   (b) If the item is strongly generatable, however, the TTP always succeeds, provided that the interested party behaves honestly. Strong fairness is achievable with robust dispute resolution.

4. Since we claim that forwardable items are also required to be describable, those items are better-suited as second items. Item validation for describable items is often simpler to perform than for indescribable ones. However, since they are also idempotent, Statement 1 may apply – unless when omitted in Table 4.1.

5. Revocable items make good first items, since they can be invalidated by a TTP if something goes wrong after their delivery (such as the second item being intentionally kept by a malicious Q). This fact makes them particularly interesting for exchanges in which the second item may be problematic – such as indescribable items, for instance.

   (a) If the item is only weakly revocable, the TTP might fail in invalidating the item. Therefore, only weak fairness is guaranteed.

   (b) If the item is strongly revocable instead, the TTP always succeeds, provided that the sender behaved honestly. Strong fairness is achievable with robust dispute resolution.

6. Co-dependent items only make sense in pairs and, as such, this fact alone may help to ensure fairness to both parties. Since they have their value linked to each other,

usually strong fair exchange can be accomplished even with minimalistic protocol design.

7. As discussed in Section 2.5.4, when the first item is generatable and the second one is revocable, accurate trustee-based dispute resolution can be implemented as a means for enforcing fairness for exceptional transactions.

In particular, item-aware protocol design focuses on how item properties would interact when two items i_P and i_Q were to be exchanged as proposed by Asokan and later authors. As we can see, when digital items are not regarded as generic objects, much information can be gained from a thorough analysis of their inherent aspects. We shall further illustrate this claim in Section 4.3, by providing an example of item-aware protocol design for a hypothetical real-world application.

It is important to notice that, in this approach, the order in which items are to be exchanged matters. For instance, the claim that revocable items are suitable for being released first in transactions concerning them; such items behave much like physical products, which can be returned to stores in situations where the buyer is not satisfied with the purchase. Therefore, revocability overcomes the difficulties introduced by the idempotency property shared by most digital items, which could result in a party keeping functional copies of a possibly unsatisfying item. For the same reason, embedding revocability into digital items seems to be a good solution for exchanges in which that same item is also idempotent.

Only recently effective item validation methods for indescribable items began to emerge [19, 63, 61], which makes most previously proposed fair exchange protocols unsuitable for them – unless indescribability is addressed somehow. Even when such items are exchanged for revocable items, no guarantee can be given to the owner against possible false-positives that might occur during item validation – as exemplified in Section 2.5.2 and illustrated in Figure 2.2. In that context, a TTP would find itself in an undecidable situation: the buyer would ask for dispute resolution, claiming that he did not receive the intended item and therefore his own item should be revoked. The seller, however, would claim that she behaved honestly and delivered the item as described – and so having the payment revoked would leave her in an unfair position. Simply demanding the seller to send the expected item to the buyer would also be unfair since, in that case, the buyer would have obtained two items and paid only for one.

Embedding some level of generatability into digital items has been the most common solution for mitigating unexpected outcomes in previously proposed protocols [68, 54, 13]. Producing generatable items is only particularly useful for fair exchange, however, if the enhanced item is intended to be released last by its possessing party; in general, there shall be no practical gain in endowing a first item with generatability – provided that a robust item validation step is implemented for that particular item – which should be taken into account when designing a new protocol.

We should notice that “perfect”fair exchange (in the sense that very few fairness violations might occur due to either party misbehavior or technical faults) might be more easily achieved when a revocable first item is exchanged by a generatable second item. Protocols designed with these particular kinds of items in mind [82] benefit from less-complicated dispute resolution subprotocols, as well as possibly more-efficient designs regarding the number of required transactions for an average successful exchange – advantages that might be lost if generic item protocol design is used instead.

4.3 A practical example of non-generic protocol design for digital items

In this section we provide an illustrative example of item-aware protocol design and how the process of designing a fair exchange protocol for a hypothetical context can benefit from taking into account inherent aspects regarding the items of interest. We conduct our example under the discussion presented in Section 4.2. We emphasize that, rather than providing a formal framework for fair exchange protocol design, our contribution provides an alternative approach to this task that is novel in the sense that it takes into account the complexity of the items being exchanged – as opposed to the conventional, arguably oversimplified approach that regards them as generic bit streams.

4.3.1 Context description and relevant items’ properties

We suppose the example protocol is meant for the electronic purchase/sale of some form of multimedia content (such as a digital audio file, for instance), and that the transaction is to be performed between two parties P (the buyer) and Q (the seller). Therefore, the items to be exchanged in the protocol are the digital payment i_P and the multimedia file i_Q.

We also assume the following properties apply for each item of interest: The payment i_P is **strongly revocable** – a reasonable assumption supported by several currently implemented third party digital payment systems [55, 11, 82, 83]. As for the multimedia content i_Q being purchased, we assume it to be both **indescribable** and **idempotent**, as supported by previously published results on the topic [19, 63, 61].

For the remainder of this section, we illustrate how the statements presented in Section 4.2 (and summarized in Table 4.1) may help the hypothetical protocol designer to take advantage of – or tackle security problems that may arise from – inherent item complexity by acknowledging these three properties.

4.3.2 Aspects that require special attention during protocol design

As suggested in Section 4.2 (Statement 6), revocable items are good choices as first items in fair exchange protocols – since they provide "step back" mechanisms to the party who is committing earlier in the protocol; this is the reason behind our choice of placing the payment as the first item to be revealed in the transaction – i.e. before i_Q is delivered by the seller. For that reason, we design our example protocol so that the payment is to be performed by the buyer during an in-transaction step with the help of a trustee-provided payment system able to enforce revocability. This approach is currently implemented in many real-world e-commerce applications and widely accepted amongst many well-known content providers (both Amazon.com and iTunes Store, for instance, offer Paypal support for their buyers).

That naturally leads to the multimedia content, which is both idempotent and indescribable, the placement as second item in the protocol. By referring to Table 4.1 we are able to conclude, from the cells that result from the intersection between the two columns corresponding to i_Q's properties and the line corresponding to i_P's revocability, that the issues raised in Statements 1 and 2 apply to our example scenario.

Statement 1 brings to our attention the fact that return policies usually do not apply for idempotent digital items – a fact that has become common practice in real-world applications that deal with digital idempotent items [9]. This creates, in our example, a context-specific requirement to take special care with how the protocol implements dispute resolution and item validation, in order to reduce the odds of a customers buying "a pig in a poke".

However, as Statement 2 emphasizes, item validation can be hard to implement for indescribable items [19, 63, 61]. For that reason, and in order to design a protocol that offers robustness against "pig in a poke" purchases, specific-context techniques for item validation may be required during protocol design. A suitable example of such technique for our instance would be the reversible degradation method [63, 61], which circumvents indescribability issues by embedding some degree of generatability (see Section 2.5.3 for remarks on generatability) to the item – while reducing the effects of non-univocal descriptions (which, as discussed in Section 2.5.2, are hard to provide for indescribable items) on the transaction.

4.3.3 Protocol suggestion

With these requirements in mind, we provide the following protocol (illustrated in Figure 4.1) as a solution for our example scenario. We assume that a previous authentication step has been performed between P and Q before the illustrated transaction takes place (which reflects the usual requirement of a buyer creating and logging into a personal account on the seller's website, for instance), in order for the transaction to take place. We also assume that the buyer has already searched seller's website for the audio file he desires to purchase, and believes it to be – based on a non-univocal description of the product – i_Q (which may be or may not be the case, since the product is indescribable [63, 61]). Also, ID_{i_Q} is the product number that identifies i_Q in Q's system. Finally, $Trans\_Num$ stands for the usual transaction label that uniquely identifies this transaction.

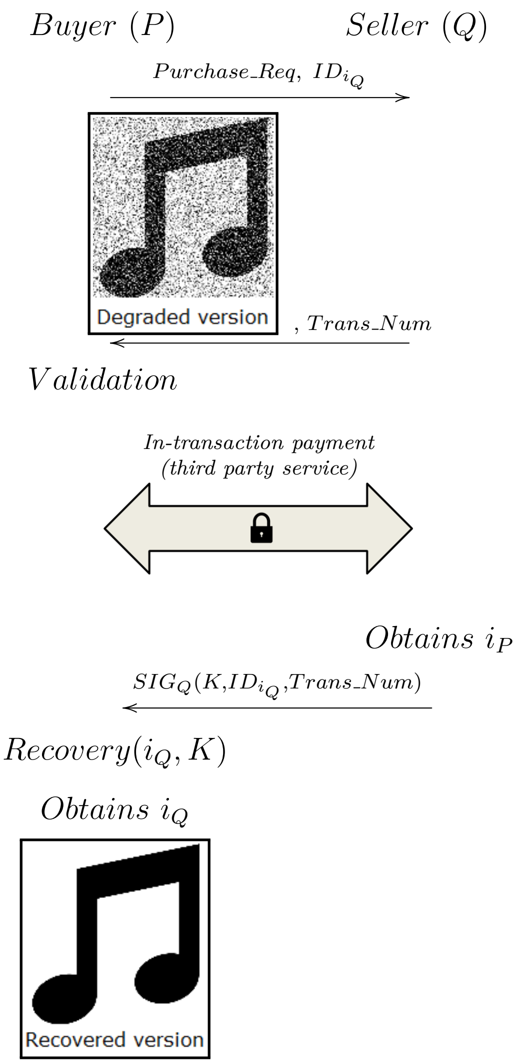


Figura 4.1: Example of item-aware protocol design in the context of digital audio purchase/sale.

As Figure 4.1 illustrates that, by acknowledging the special characteristics of the items of interest, the careful designer can more-easily focus on solving context-sensitive issues

– which might be of use in the task of avoiding common protocol design pitfalls. The suggested protocol relies on a third-party provider for the payment step (thus ensuring revocability) and on the reversible degradation method as a means of circumventing indescribability (which ultimately introduces some robustness against unsatisfactory outcomes on the behalf of the buyer and the subsequent impact of no-return policies, common to scenarios concerning idempotent digital items). Also, since some degree of generatability is also enabled by the use of reversible degradation in the validation step, dispute resolution – when required – should be easier to enforce (Statement 3).

In particular, the use of reversible degradation in this protocol allows the buyer to obtain a sufficiently degraded (i.e., worthless), but still fully-playable version of i_Q before payment – which he can then listen to in order to make sure i_Q is in fact the product he intends to pay for. If it is, the buyer proceeds with the protocol by paying for the recovering key K that will be used as input, together with the degraded version of i_Q, in the recovery process that restores i_Q to its full original quality.

If the degraded version brings the buyer to realize, however, that i_Q is not in fact the product he desires, he can simply abort the protocol without paying for (and thus without obtaining) K – which ensures fairness for the buyer. Because the full quality version of i_Q cannot be obtained from the degraded version without K, seller's fairness is also guaranteed. Exceptional outcomes for the transaction would include, for instance, situations in which $K' \neq K$ is delivered after payment – which would prevent the buyer from successfully recovering i_Q; but even in this scenario, dispute resolution would be simple to accomplish (since no ”wrong product-- only a wrong key – had been delivered, no issues concerning no-return policies apply; the judge would be able to settle the situation either by demanding the correct K from seller, or by revoking i_P).

Capítulo 5

Visual Cryptography Authentication

In this chapter we present a robust authentication method for sensitive transactions that guarantees to both parties that 1) they are in fact communicating with the intended counterpart (mutual authentication); and 2) only legitimately-initiated transactions will be committed (transaction authentication) – thus addressing well-known issues, such as the Man-in-the-Middle attack, even under hard-to-tackle scenarios as the ones foreseen by the modern adversary model [56]. Our solution relies on a novel approach towards Visual Cryptography (VC) [52], designed with the purpose of overcoming previously-known limitations [52, 51, 86, 28] regarding the use of VC in authentication scenarios.

The remainder of this chapter is organized as follows: Section 5.1 introduces the context to which our method applies. Section 5.2 describes our main motivation and application scenario, focusing on the definition of the modern adversary which we intend to defeat. Section 5.3 presents relevant related work to our discussion, including a brief overview of other solutions currently used in the context of e-banking. In Section 5.4 we introduce our proposed solution and describe its main components. The building blocks for the critical component of our solution are further detailed in Section 5.5; these building blocks are then combined in Section 5.6 in order to produce an implementable method for our solution.

5.1 Introduction

Visual Cryptography (VC) was proposed by Naor and Shamir [52] as a secret sharing cryptographic paradigm that combines perfect ciphers and perceptual information recovery. The general k-out-of-n VC procedure works as follows: Initially, a secret image is split into n raster images that, alone, reveal no information about the secret. Each share is then delivered to a user. The original secret is retrievable only if k or more of those shares are stacked together, so that no smaller subgroup of users can find out the hidden information. Since the recovery does not require any computation – relying only on the perceptual capabilities of the users instead – VC schemes provide a simple and intuitive alternative for secret sharing among two or more parties, which is of particular interest in contexts where sensitive information is to be provided to users with little or no technical knowledge, or with little or no computational power.

As Internet and mobile banking (or simply *e-banking* and *m-banking*, respectively) becomes increasingly popular among customers, their potential as targets for fraud has also been noticed to increase [2]. In the recent years, attacks to e-banking applications have become more complex and serious threats empowered with malicious software artifacts (*malware*), that explore not only security breaches on the e-banking implementations itself, but also user misinformation. Once the adversary manages to get an instance of *malware* running in the victim's device, ensuring any level of security becomes a hard – not to say impossible – task: the perpetrator becomes able, for instance, to observe incoming and outcoming information in the infected device; to record user inputs (through the keyboard, mouse or any other currently available input device) and report them to a remote server; to forge screens to be displayed to the user in runtime.

In fact, the current state-of-the-art of *malware* allows the infected device to be controlled to such an extent as to enable the adversary with the power of undetectably impersonating the user to any party willing to communicate with him (a bank server, for instance) and/or vice and versa, thus becoming able to manipulate user-initiated transactions even when several layers of authentication are implemented by the system. This extreme type of attack – known as the *Man-in-the-Middle* (MitM) attack – has become as feasible as it is dangerous in the past few years, and remains the focus of current related research [26, 41, 33, 32, 56, 21].

To the extent of our knowledge, no currently-available approach is able to effectively undermine MitM attacks under the modern adversary model. The most successful approaches usually make use of multi-factor [17, 7] or multi-channel authentication mechanisms [32]. The first rely on secondary devices/communication channels in order to extend the authentication protocol to the outside of the hypothetically compromised terminal – thus adding an extra layer of security to the authentication system. The latter, on the other hand, strengthens the accuracy of the authentication process by using at least two out of three *authentication factors* as obstacles to the adversary: password authentication, for instance, relies on the first factor – *"something the user knows"*; SMS confirmation codes for PC-initiated transactions are intended to verify the user's identity according to the second factor – *"something the user has"*; and finally, biometric authentication systems [23, 84] are based on *"something the user is"*.

Our two-factor/two-channel authentication solution is based on Visual Cryptography (VC) [52] and illustrated with (but not limited to) the e-banking scenario. It is novel in the following aspects: 1) it does not rely on any assumptions regarding uncompromised devices; 2) it satisfies both two-party and transaction authentication requirements [20] without storing credentials inside any user-side (compromisable) device; 3) it is cost-effective in comparison to currently implemented solutions; 4) it is designed to be robust even in the realistic scenario where a user's device is hijacked by *malware* – an increasingly realistic possibility [2]; and 5) it effectively protects user e-banking transactions against both *malware*-oriented attacks (such as MitM and credential stealing, for instance) and certain kinds of social engineering attacks (such as *phishing*). We present our solution in detail and in a step-by-step, constructive approach, also providing a comparison with state-of-the-art solutions regarding security, usability and logistical aspects.

5.2 Motivation and application scenario

In this section we further describe the context to which our proposed solution applies. We summarize previously published discussions about the modern adversary capabilities and provide details on why it has become inaccurate to rely on the classical adversary model when proposing practical e-banking solutions.

5.2.1 Context description and basic definitions

Most current e-banking applications rely, at least partially, on login/password based authentication for establishing secure sessions between $User$ (that is, the bank account holder) and *Bank* (that is, the financial institution server where the e-banking application is implemented). Because passwords can be easily stolen in the real-world scenario (as we shall see in Section 5.2.2), two-factor/two-channel authentication protocols have become common practice [26, 41]. Such approaches usually rely on some form of login/password request as the initial authentication step and, whenever a sensitive transaction is requested by $User$, a second authentication step – intended not only to double check $User$'s identity, but also to ensure that the parameters entered for the transaction (destination account number, amount, etc) were not tampered with by any malicious party before reaching *Bank* – takes place. This second step, usually referred to as *transaction authentication* (as opposed to *user authentication*) [20], has recently become a solid requirement for practical e-banking solutions designed to be robust against MitM attacks and has been recognized as a crucial aspect for future e-banking security [56]. For the remainder of this chapter, we shall rely on the following definitions:

- **User authentication**: A solution satisfies the user authentication requirement if it produces unequivocal evidence to at least one of the involved parties about the identity of the other party. Specifically, if both parties receive evidence about their counterparts' identities, the solution is said to provide *mutual user authentication*.

- **Transaction authentication**: A solution satisfies the transaction authentication requirement if it produces tamper-proof evidence containing both the transaction parameters and freshly-generated information, that can only be produced by the involved parties and retrieved by those parties. Specifically, one of the parties (usually the bank) is able to produce a transaction authentication artifact[1] that can only be retrieved by the other party (usually the user); and upon a successful retrieval, the user receives the guarantee that the authentication artifact was in fact originated by the bank and was not tampered with by any adversary.

Several well-known approaches can currently be used in the implementation of second-factor and/or transaction authentication, such as transaction authentication number tables (TANs), secure tokens, mobile communication devices, to mention a few. The effectiveness of such methods greatly vary in security, efficiency, cost and usability[2], and all of them aim at making e-banking authentication more robust against the modern adversary.

For the remainder of this section, we summarize previously-published results in order to detail how the modern adversary deviates from the classic adversary model, and as such is capable of overcoming the traditional use of cryptographic artifacts.

5.2.2 The modern adversary in the current Internet banking context

In the classical Dolev-Yao adversarial model [24], the adversary is assumed to control the communication channel established between parties – thus being able to passively or actively attack the messages sent back and forth, but unable to compromise the channel's end points. However, recent state-of-the-art *malware* design, added to user misconception about the current available virtual threats, empowers the modern adversary with peer control capabilities; this means that in today's scenario, not only the communication channel, but also computation performed inside the peers' devices themselves (specifically, the account holder's PC, in our scenario), can be monitored and tampered with by malicious third parties [56].

Once a *malware* instance is installed in the victim's device, several unauthorized operations can be remotely and silently executed by the intruder in order to mount several forms

[1] For simplicity of explanation, we shall refer to this information simply as authentication artifact for the remainder of this paper.

[2] We shall provide a brief discussion, for comparison purposes, regarding the most prominent alternatives in Section 5.3; more-detailed studies can be found in previously published work on the subject [32, 35, 33]

of attacks against e-banking applications. Three classes of attacks have been distinguished in previous research accounting such modern adversary model: credential-stealing attacks, channel-breaking attacks and content-manipulation attacks [56].

Credential-stealing attacks refer to attempts that the adversary might perform in order to obtain user credentials – either by extracting them from the compromised device where they are stored, or by tricking users into revealing them through social engineering or *phishing* techniques. This kind of attack is aimed at obtaining the necessary information for successfully impersonating the user in future transactions and, as such, are also referred to as *offline* attacks.

Channel-breaking attacks, on the other hand, are intended to break the hypothetically secure communication channel established between peers. Such attacks include the well-known Man-in-the-Middle (MitM), in which the adversary acts as a transparent proxy peer between $User$ and *Bank* by maintaining two simultaneous authenticated sessions (one with $User$, pretending to be *Bank*; and another one with *Bank*, pretending to be $User$). In order to accomplish that, the adversary can either try to convince $User$ to accept an invalid e-banking certificate, or even not use any certificate at all – as many users do not usually check certificates before accepting authenticated connections [56]. As opposed to credential-stealing, these attacks are mounted in real time and therefore also known as *online* attacks.

A much more subtle and worrisome form of perpetration can be found in the so-called **content-manipulation attacks**. These threats can effectively undermine most known transaction authentication techniques by deceiving $User$ into providing *Bank* with confirmation over bogus transactions tampered with by the adversary. Such attacks include, for instance, the ability to replace original transaction parameters (for instance, the destination account number for a user-initiated money transfer could be replaced with the adversary's account number) and the ability to replace confirmation content originated by *Bank*, meant to be displayed to $User$ through the contaminated device's display.

It should be noted that these three classes of attacks are by no means exclusive and can be (as they usually are) combined by the adversary in order to successfully defeat e-banking security. In Section 5.3 we proceed to analyze some of the most commonly found solutions, regarded as the current state-of-the-art in e-banking security, and discuss their main advantages and weaknesses regarding the modern adversary capabilities we just described.

5.3 Related work

In order to minimize the threat posed by an adversary capable of compromising client devices, several security techniques are currently implemented by e-banking providers.

Much research has been done on the effectiveness of such techniques for the prevention of the attacks described in Section 5.2 and, specifically, the MitM attack. In this section we present a brief summary of their conclusions.

5.3.1 Two-factor and two-channel authentication

Solutions for authentication based on the classical adversary model usually rely on authentication protocols (such as SSL/TLS) [35] for establishing a secure channel between peers for further communication of sensitive transaction data. Such approaches often include only one authentication factor (a long-term, previously established password, for instance) and are not effective against an adversary with the capabilities described in Section 5.2: by observing a single session in a one-factor authentication protocol, an adversary with credential-stealing capabilities can easily obtain $User$'s credentials (in this case, his login and password) and impersonate him in future, adversary-initiated sessions. Furthermore, an adversary with channel-breaking and content-manipulation capabilities is able to tamper with the parameters of a transaction initiated by $User$ in order to trick $Bank$ into sending, for instance, a different amount of money to a different destination bank account.

For that reason, e-banking authentication has been deemed to require *two-factor authentication* protocols [56]. The main idea of such methods is to require an initial, often password-based user authentication round in the beginning of each session, and a subsequent stronger, more robust user authentication round whenever a sensitive transaction is requested in the established session (a money transfer, for instance); in some approaches, the second user authentication round may also include transaction authentication steps – an attempt by $Bank$ to make sure that the received transaction parameters have not been changed by a content-manipulation-capable adversary.

It has been noted that, even though two-factor authentication can increase robustness against several modern adversary threats, it cannot be much more effective than one-factor authentication if the second round is performed on $User$'s primary (possibly compromised) device [26, 32, 56, 33]. For that reason, *two-factor, two-channel authentication* methods [32] have become widely adopted in the financial market. Such approaches rely on requiring a second external, often offline and/or tamper-resistant device in the authentication protocol; because such secondary devices are independent from $User$'s primary (possibly compromised) device, security is enhanced.

5.3.2 One-time passwords (OTPs) and transaction authentication number (TAN) tables

One well-known approach for providing user two-factor authentication is the use of one-time passwords (OTPs). Such protocols rely on randomly-generated authentication numbers that are used only once, in a challenge-response-based scheme. These numbers can be either generated at transaction time by synchronized devices (i.e., OTP generators) or provided to $User$ beforehand by regular letter as a paper-based table (the so-called transaction authentication number (TAN) tables) [56]. Since any given OTP is supposed to be used only once by $User$, recording it for future use would be of little help to the adversary.

However, even though OTPs can, in the event of $User$'s long-term password being stolen, make future adversary-initiated transactions harder to perform, they fail to provide transaction authentication. Specifically, OTP-based methods alone are known to be ineffective against MitM attacks, for $User$ can be led to authenticate a transaction that has been tampered with by an adversary in runtime [56, 32]. In other words, OTPs effectively mitigate offline attacks, but are not robust against online and content-manipulation attacks (as described in Section 5.2.2).

Another disadvantage of TAN tables is that they introduce an arguably small degree of inconvenience for $User$, since one needs to carry their TAN table with them at all times if they want to be able to perform e-banking transactions on-the-go. More than inconvenience itself, the problem lies on how easily $User$ can, for convenience sake, simply scan his TAN table (or simply copy its contents into a text file) and store it in digital form in his adversary-controlled device – thus allowing the intruder to obtain the digital copy and further mitigating any security enhancements provided by the solution.

5.3.3 Cryptographic tokens

Another approach for minimizing the effects of a compromised device on authentication is to use PKI infrastructure with externally-stored credentials. *Cryptographic tokens* [35] are tamper-resistant devices that can be connected to $User$'s PC (usually through a standard USB interface) during the transaction, upon *Bank*'s request, and that can contain a pair of public and private keys. These devices are designed to disallow their contents to be copied by external applications (i.e., the adversary), and can be embedded with computation capabilities (encryption/decryption, signing/verifying).

Tokens are currently available in variable fashion – from simple USB storage-like looking devices to full-featured screen/keyboard-equipped gadgets. In the screen-equipped variant, $User$ can securely verify the transaction's parameters before signing it – since the adversary's content-manipulation skills are limited to the victim's compromised PC

– and therefore transaction authentication can be ensured. If not, however, *User* has no control over what is being signed/encrypted within the token; in such cases, the adversary might still be able to replace the information sent by *Bank* for token authentication and succeed in altering the transaction.

Unfortunately, token devices remain relatively too expensive [32] to be widely implemented in a one-per-customer basis – specially the more-secure variant equipped with input or output extensions for transaction authentication. Also, in the event of a lost or compromised token, the credential revocation/reissue can lead to even larger costs; furthermore, due to the token being tamper-proof by design, any form of update is unpractical.

5.3.4 Mobile communication devices

One approach that has recently become a trend in e-banking research is the use of mobile devices (specifically, smartphones) as secondary devices for two-factor, two-channel authentication [26, 41, 66, 7]. The fact that such devices have become so common among the average consumer makes such methods practical and relatively low-cost to implement. Furthermore, because mobile devices are equipped with input (keyboard, touchscreen) and output (display) capabilities, they also become an attractive option for fulfilling transaction authentication requirements.

However, the exact same reason that makes mobile devices so attractive for e-banking systems is also what makes them an obvious next-in-line target for the modern adversary: their widespread adoption among the public and multiple purpose orientation. In fact, mobile *malware* findings have become extensively common in the most popular systems and is already considered an increasingly serious threat [80, 27]. To make things worse, the allegation that mobile devices are completely independent from the user's primary device is inaccurate; first because most users usually connect their mobile devices to their (possibly contaminated) PCs by means of standard interfaces such as USB or Bluetooth; and second, because a user that is inclined to unsafe online behavior is likely to reproduce that behavior regardless of the nature of his terminal (PC, smartphone or otherwise). This makes it possible for the adversary to deliver mobile-specialized *malware* directly from the primary device into the secondary one, thus obtaining control over the two channels used for authentication and becoming hypothetically capable of performing a synchronized Man-in-the-Middle attack in both.

If we consider the likely scenario of an adversary able to first infect *User*'s PC and then, upon detecting a mobile device connection to the PC, to inject system-specific *malware* on that device, it becomes clear that all forms of attack discussed in Section 5.2 can be extended to the mobile device as well. In that case, storing user credentials on the mobile device becomes as dangerous (and ineffective) as storing them in *User*'s PC – since they can then be stolen in the exact same way through credential-stealing attacks. Channel-breaking and content-manipulation attacks also become easy to perform, provided that the adversary has obtained control over the second channel and has become once again able to effectively perform MitM attacks.

For all these reasons, we firmly believe that future methods relying on mobile devices as secondary channels for two-factor/transaction authentication should be proposed taking into account the ongoing increase in mobile *malware* development. We are currently unaware of any mobile-based e-banking solution that remains robust under the assumption of a compromised device, which our proposed method (see Section 5.4) aims to provide.

5.4 A Visual Cryptography solution robust against mobile *malware*

In this section we propose a novel user/transaction authentication technique for enabling two-factor, two-channel e-banking based on Visual Cryptography (VC) [52]. Our solution is intended to be robust against the modern adversary described in Section 5.2, and designed to be cost-efficient and effective in scenarios in which both primary (PC) and secondary (mobile phone equipped with high resolution display) devices are possibly compromised by *malware* and, as such, controlled by an adversary.

5.4.1 Concept description

As discussed in Section 5.3.4, mobile devices have been noted to be convenient, relatively cost-efficient platforms for the implementation of e-banking two-factor, two-channel authentication solutions. In order to simultaneously provide mutual user authentication and transaction authentication, our solution relies on *User*'s mobile device mostly for its output characteristics – as opposed to relying on its computation capabilities, as done in previously proposed mobile e-banking solutions [26, 41]. Specifically, we regard the mobile device as being capable of receiving and displaying tamper-resistant, VC-encrypted and transaction-dependent information from *Bank* that can only be recovered by *User*, through the possession of a previously-established shared secret known otherwise only by *Bank*. This shared secret takes the form of a physical, transparent **VC-card** generated by *Bank* and delivered to *User* upon account opening, much like *User*'s ATM card. We shall describe how the solution makes use of this shared information in Section 5.4.4.

Since VC decryption is performed by the user's eyes, it requires no sensitive computation whatsoever. Also, the shared secret between *User* and *Bank* is not stored in any *malware*-compromisable device (in fact, the VC-card is designed to be stored in *User*'s wallet or pocket, for instance – much like the standard ATM banking card). All these makes it impossible for the modern adversary to perform credential stealing attacks through *malware*-infection. Also, the tamper-proof nature of the produced authentication artifact (or *authentication share*, in the context of our solution) overcomes the adversary's content-manipulation capabilities. Finally, since the produced authentication shares can only be generated and used to reveal information to parties who know the shared secret, and because this secret is both impossible for $User$ to memorize, and hard to replicate (physically and digitally) without losing its functionality, $User$ is unable (or at least unlikely) to (un)intentionally reveal the secret to a third party; this makes the solution able to overcome channel-breaking issues.

5.4.2 Classical Visual Cryptography

The original concept of Visual Cryptography (VC) [52] was proposed as an alternative method for secret sharing between parties who could not rely, due to limited or inexistent resources, on traditional computationally-based systems. The main idea behind the classic method is to convert the secret into a bitmap, black-and-white, low-resolution image and resize it in such a way that each pixel (used interchangeably with **subpixel** from now on) becomes represented by a square matrix of pixels (further referred to as **superpixel**); the image is then subdivided into a predefined number of seemingly randomly-generated **shares** (in our context of application, we consider two shares).

In the specific two-share application of VC, for instance, the secret image is initially extended by a factor of 2 – which results in an extended secret composed by 2 × 2 superpixels formed either by four black subpixels (aka, an **information** superpixel) or four white subpixels (aka, a **non-information** superpixel). The pair of shares for recovery is then constructed in the following fashion: First, 2 × 2 superpixels composed of two white and two black subpixels are randomly-selected from the six possible patterns and included in the first share (or **secret share**, in the context of our solution). Finally, the superpixels for the second share (or **authentication share**, in the context of our solution) are chosen to be either identical to the equivalent in the first share (if the equivalent in the extended secret is a non-information superpixel) or complementary to it (if the equivalent in the extended secret is an information superpixel). Figure 5.1 illustrates all the possible superpixel choices for composing first and second shares in the two-share instance of original VC.

After the pair of shares is constructed, each share is finally given to one of the interested parties, and the recovery of the original secret can only be performed if both shares are presented and stacked together, one on top of the other, in no particular order. We refer

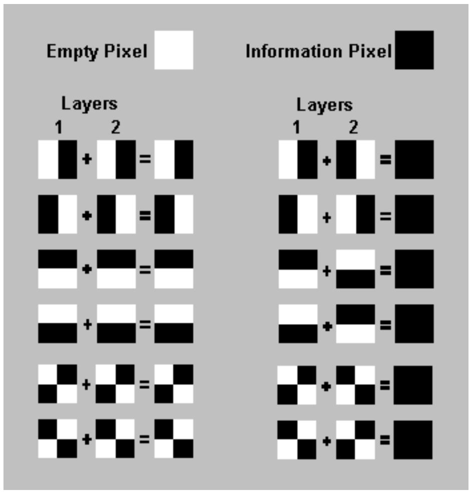
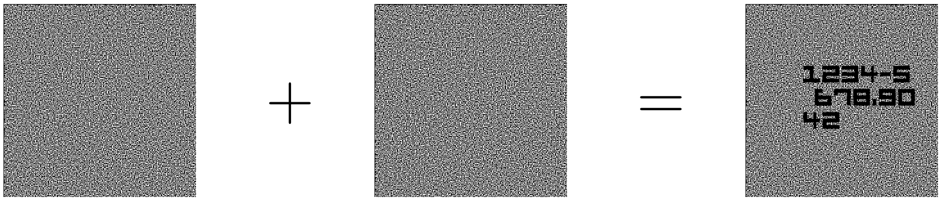


Figura 5.1: Superpixel choice for shares construction in the two-share classical VC scheme.

to this process as **overlaying**.[3]. Figure 5.2 illustrates secret recovery by a party that is presented with her counterpart's share.



First share

Second share

Overlaid shares

Figura 5.2: Secret recovery by overlaying a pair of shares in the two-share classical VC scheme.

As we can see in Figure 5.2, non-information superpixels in the overlay are recovered in shades of gray, while information superpixels appear in solid black. This "black and gray" aspect of the recovered secret accounts for a 50% contrast loss in comparison to the original extended (black and white) secret, which is not enough to undermine image visualization.

Considerations on the original VC method

The original VC method has two main limitations that concern our intended application scenario: First, each pair of shares can only reveal one predefined secret. This means

[3] It should be noted that, in order to allow recovery, at least one share has to be embedded with transparency for white subpixels. For that reason, we shall use the terms share and transparency interchangeably from now on.

that every interaction regarding the sharing of a particular secret between two parties requires the generation and distribution of a new pair of shares. Further research on VC addressed this inconvenience with the proposal of multi-secret Visual Cryptography (MSVC) methods capable of sharing more than one secret within the same collection of shares [86, 28], by exploring several rotation angles in which shares could be overlaid. However, such approaches still require all the intended secrets to be known beforehand, since they are used as input for the shares generation method.

Second, straightforwardly reusing any previously-generated share (specifically, the first share) in a new pair intended to recover a different secret is known to be insecure [52, 51]. For instance, in a scenario in which a particular first share would be reused in the construction of several different second shares, intended for the recovery of several different secrets, even a classical adversary would be capable of reconstructing the first share almost entirely by observing no more than two transactions (considering that each transaction includes the transmission of a different second share through an insecure channel); the reconstruction method in this context is well-known and follows from the one-time-pad aspect of classical VC. This poses a logistic issue on the application of classical VC to practical authentication scenarios – since a party would not be able to authenticate more than one transaction with one given transparency.

Even though rotation-based MSVC methods proved the possibility of sharing more than one predefined secret within a fixed set of shares, few VC methods have been proposed in order to allow secure transparency reusability in sharing multiple non-predefined secrets [51]. If the secrets intended to be shared are known to be sufficiently smaller than the transparency intended to be reused, for instance, that transparency can be subdivided into several disjunctive regions (aka, *cells*), which can then be interpreted as a collection of independent, smaller first shares. The authentication share generation process would then keep track of which cells have already been used in past transactions and, upon a new authentication iteration, would randomly choose a previously unused cell as the input – together with the new secret image containing the transaction parameters for authentication – for a new authentication share construction. Since this newly-generated authentication share would contain no information about previously-used cells from the first share, a malicious observer would not be able to use it for comparison with previously-collected authentication shares, thus effectively allowing the reusable transparency to be securely applied for recovering multiple secrets.

This approach holds, however, little practical applicability for real-world scenarios. First, it is obvious that the same level of reusability would be achieved if, instead of providing $User$ with a transparency that is large enough to be subdivided, $Bank$ provided $User$ with a block of smaller, cell-sized transparencies in the first place. $User$ would then be instructed to sequentially use each transparency once and then discard it every time a new transaction was performed, an approach in many ways similar to TAN tables (see Section 5.3.2). On the other hand, comparatively to TAN tables, while those are able to provide at most user authentication, the cells approach are able to include transaction-dependent information in the protocol – thus enabling transaction authentication as well.

Nevertheless, it would be convenient if the level of reusability of transparencies could be enhanced, thus reducing reissuing costs. For the remainder of this section, we describe a novel approach to VC authentication intended to significantly enhance the reusability level of *User*'s transparency securely. Specifically, our method allows previously-used regions to be selected more than once without enabling first share reconstruction attacks before the transparency needs to be replaced. In practical terms, this means that party *Bank* can issue a shared secret share to party *User*, which will then be used to authenticate a reasonable number of future e-banking transactions while effectively guaranteeing the following requirements: 1) *Bank* knows that only *User* can retrieve each of the secrets and complete each authentication iteration; 2) the decryption process does not rely on any of *User*'s devices (primary or secondary) to be uncompromised; 3) *User* knows that only *Bank* could have been the originator of any authentication share that reveals coherent information when overlaid with his secret share; and 4) the transaction about to be committed is indeed the one intended by *User* and has not been tampered with by any adversary, provided that his abilities fit the model described in Section 5.2.2. We notice that, by doing so, our method enables the protocol to provide both mutual user authentication (items 1) and 3)) and transaction authentication (item 4)) – even if the used devices are compromised by *malware* and controlled by an adversary (item 2)).

5.4.3 Components description

We initially define our solution to be composed by the following methods, which are to be executed by *Bank* as we shall illustrate in Section 5.4.4:

M1. **SecretShareGen()**:

- *Input*: extension_factor, random_seed
- *Return*: secret_share

M2. **SecretGen()**:

- *Input*: secret_data, extension_factor
- *Return*: exended_secret

M3. **AuthShareGen()**:

- *Input*: secret_share, extension_factor, extended_secret
- *Return*: authentication_share

In method $M1$, a new random black and white VC share is generated. This is performed by randomly selecting superpixels from all the possible patterns (given by *extension_factor*), in a similar fashion to previous VC methods [52, 51]. Method $M2$ simply takes a string or image *secret_data*, embeds it as black visual data into a white background image and resizes it to match *secret_share*'s dimensions by expanding each pixel into an $extension\_factor \times extension\_factor$ superpixel. Finally, method $M3$ outputs an authentication share that, when overlaid with *secret_share*, reveals *extended_secret* with arbitrary contrast loss. Because we intend the same instance of *secret_share* to be used in the generation of multiple instances *authentication_share* in a secure fashion, the proposed method $M3$ is where we part ways with previous VC techniques; we shall detail $M3$ in Section 5.5. For the remainder of this section, we describe how the proposed solution can be applied in the context of two-factor, two-channel e-banking authentication.

5.4.4 Protocol description

Upon opening a new bank account for a new customer $User$, *Bank* generates a new random secret share and prints it as a physical, transparent VC card – which is then delivered to $User$ in person or by regular mail; also, *Bank* stores the digital version of $User$'s transparency in his secured server, together with $User$'s account data. The secret $User$ share is generated by *Bank* with method $M1$, which can be repeated every time a new VC card has to be reissued (if the previous transparency has been lost/compromised or used in enough transactions to be considered insecure for future authentications). We note that $User$'s transparency is obtained similarly to previous VC proposals – that is, by randomly selecting $(extension\_factor)^2$-sized superpixels from all possible patterns; we shall discuss the effects of allowing variable *extension_factor* values in Section 5.5.3.

After acquiring his VC card, $User$ becomes able to engage in e-banking transactions with *Bank*. A two-factor, two-channel protocol example that employs our proposed solution is illustrated in Figure 5.3. As described in Section 5.3.1, the proposed solution is intended as a second factor/transaction authentication technique, and as such we assume that an e-banking session with *Bank* will be initiated by $User$ in some other way (for instance, by using his login and long-term password in the first authentication round, to be performed through his primary device's web browser) in order to grant $User$ access to perform specific transactions. This first authentication round is independent of our proposed solution and is represented in *Step* 1.

After a new session with *Bank* has been established, $User$ proceeds to initiate the intended transaction. Let us suppose that $User$ intends to perform a money transfer of

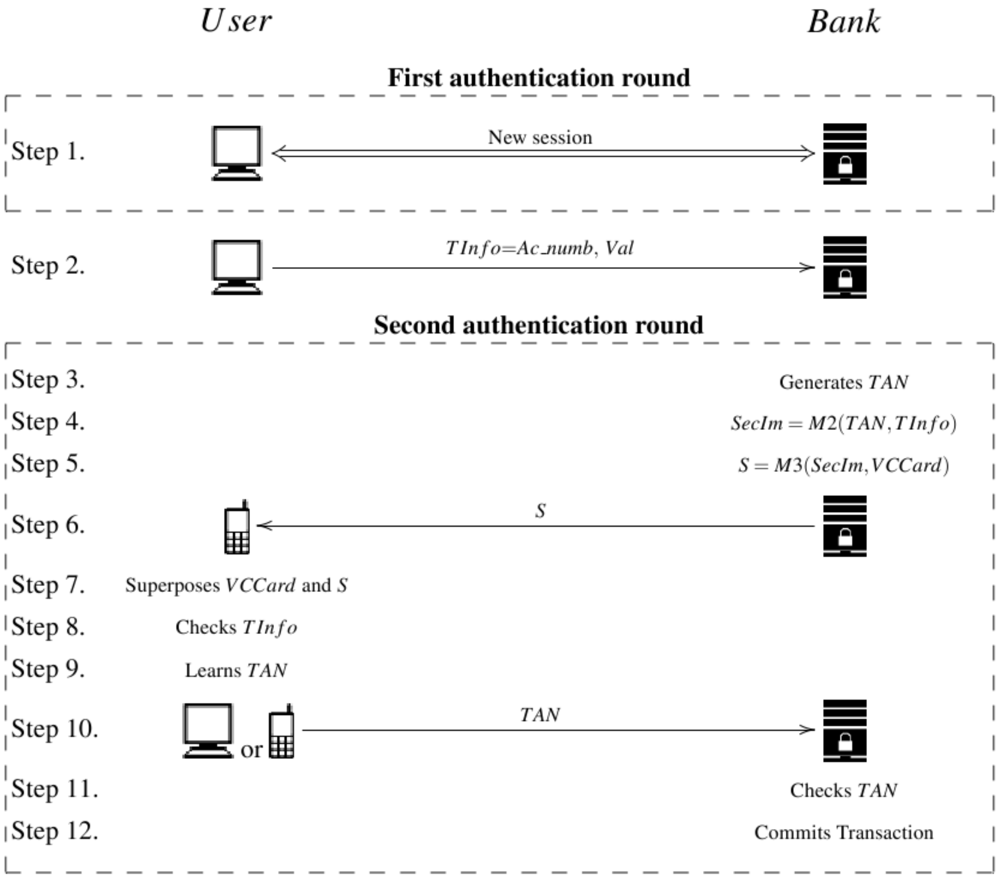


Figura 5.3: Two-factor, two-channel authentication protocol for Internet banking with the proposed solution.

value Val to destine account *Ac_numb*; these parameters are informed to *Bank* through $User$'s primary device (which we assume to be his PC) in *Step* 2., and constitute the transaction information $TInfo$.

Upon receiving the money transfer request, *Bank* puts the transaction on hold and begins the second authentication round. The requested transfer will only be executed if and when *Bank* verifies that 1) it was in fact initiated by $User$ (user authentication) and 2) the received $TInfo$ has not been altered by any malicious party before reaching *Bank* (transaction authentication). In order to accomplish that, *Bank* generates a random transaction authentication number TAN (*Step* 3.) and uses it as input to method $M2$, together with $TInfo$ – thus obtaining a black-and-white extended secret image *SecIm* (*Step* 4.).

Bank then retrieves the digital copy of $User$'s VC card and uses it, together with

$SecIm$, as input to method $M3$ in order to obtain a transaction-specific authentication share S (*Step* 5.). We notice that, in order to allow the same first share (VC Card) to be securely used in the authentication of multiple transactions, method $M3$ – that is, the method for generating second shares for authentication with the VC Card by $User$ – remains the most delicate process of our solution; we describe method $M3$ in detail in Section 5.5.

Once *Bank* obtains S, he sends this information to $User$'s mobile device (*Step* 6.). Upon receiving image file S, $User$ displays it on his mobile device and places his transparent, physical VC Card on the screen – carefully adjusting it to match the position of S (*Step* 7.). Once the two shares are perfectly aligned, the original information $SecIm$ can be visualized by $User$: he checks whether the transaction parameters $TInfo$ are correct (*Step* 8.) and learns the transaction authentication challenge TAN (*Step* 9.). If the revealed information regards the transaction initiated by $User$, he finally responds to the challenge by sending TAN to *Bank* (*Step* 10.); this can be done either from his PC or mobile device, depending on how the application is implemented. Upon receiving the correct TAN (*Step* 11.), *Bank* can safely assume that the transaction was in fact initiated by $User$ – since only he has the required VC Card which enables him to learn the secret TAN – and thus can be committed (*Step* 12.).

5.5 Second share generation method description: trading contrast for reusability

In order to provide the security requirements of mutual user and transaction authentication with Visual Cryptography, we first need to be able to circumvent the one-time-pad aspects described in Section 5.4.2. In other words, *Bank*'s authentication shares have to be generated in such a way that the adversary is unable to reconstruct $User$'s VC card by pairwise comparison of previously-collected authentication shares – even when a reasonable number of authentication shares has been collected for analysis.

To accomplish that, we revisit the definitions of information and non-information superpixels in an overlaid pair of VC shares, which we introduced in Section 5.4.2). For the remainder of this section, we present several approaches to generate multiple authentication shares suitable for being superimposed with a fixed, reusable secret share. Since the presented approaches are not mutually exclusive, we do not consider them to be individual methods for authentication share generation; instead, we regard them as the building blocks that, ultimately, will be progressively combined for a robust authentication share generation method against reconstruction attacks; the final method can be found in Section 5.6.

5.5.1 Contrast vs. reusability

In Section 5.4.2, we outlined the central idea behind the original VC proposal – which is to decompose a secret black and white image into a pair of shares, and to enhance pixel entropy on each share so that neither of them can, by itself, reveal any information about the secret within. The main idea behind VC methods is, therefore, to sacrifice some degree of contrast (from recovered secret image, when both shares are overlaid) in favor of security.

This loss of contrast comes from how the superpixels composing the second share are selected: either the same superpixel as the one chosen for the first share is included (if the correspondent secret superpixel was a non-information/white one) or the first share's superpixel's complement is included (if the correspondent secret superpixel was an information/black one); this procedure is illustrated in Figure 5.1. By selecting superpixels in that fashion, while information superpixels remain completely black in the recovered secret, the originally-white non-information superpixels become in fact grey – which causes a 50% contrast loss in the recovered secret in comparison to the original image.

We note that, while this fixed 50% contrast loss introduced by the original VC method prevents each individual share from revealing the original secret, it is also the reason behind the one-time-pad characteristics of original VC – which accounts for why transparencies cannot be effectively reused. In other words, while each superpixel composing the first share is randomly-chosen from six possible "two black, two white" patterns, each choice for the second share is virtually limited to two possibilities (either the same one, or its complement).

Therefore, if the same first share was used for the construction of two second shares intended for the recovery of two different secrets, simply collecting both second shares upon transmission and pairwise comparing them would be enough for the adversary to trivially determine the correspondents, in the first share, of all the superpixels that are not simultaneously used for information in both second shares[4].

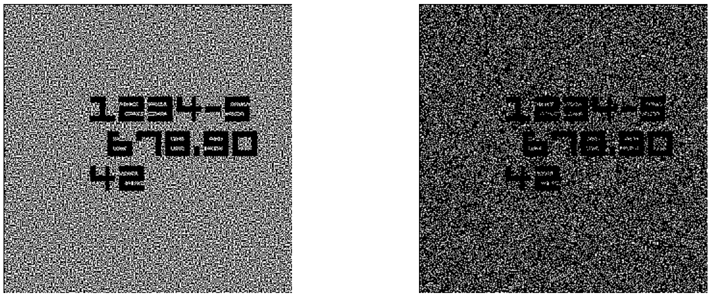
By allowing non-information superpixels to be chosen randomly - in the same fashion as the choice made for the first share – we are able to eliminate the limitation that these superpixels must be identical on the first and second shares. This means that, supposing that an adversary recovers multiple second shares originated from the same first share, it would be harder to determine the value of the correspondent superpixel in the first share – even under the assumption that most of the superpixels will be non-information ones. In this case, the knowledge obtained from simple pairwise comparison of second shares would be severely limited to regions of information intersection – that is, groups of

[4] This statistical recovery-by-comparison relies on the fact that most of the superpixels in each share refer to non-information superpixels in the recovered secret and, as such, are identical to their correspondents in the first share.

information superpixels situated in the same coordinates on both second shares. This is true because, even though we introduced some freedom in the choice of non-information superpixels composing the second share, the information superpixels in the first share still have only one possible complement. Since such information intersections (when present) are likely to be formed by several neighbor superpixels (depending on font size and shape, for instance), these regions would represent groups of neighbor superpixels with identical patterns in both second shares.

However, we notice that in our context of application, the amount of information superpixels in any recovered secret is significantly smaller than that of non-information superpixels – thus leading to a statistically small percentage of information intersection superpixels. For instance, if the number of superpixels used for information in a second share account for 10% of the total number of superpixels, the average intersection area between two previously observed second shares would be no more than 1% their size. Therefore, in this scenario, the trivial recovery of first share by pairwise comparison becomes ineffective.

Another aspect to be taken into account is that enhancing the contrast loss in the recovered secret means making it harder for $User$ to visually recover the secret information by overlaying both shares. Figure 5.4 illustrates how much secret recovery is affected by the method described in this section. We consider the final quality of the recovered secret to be too poor for practical application – and thus do not recommend this method alone to be used as the sole component for embedding reusability in VC. Instead, we shall proceed to use it as the main building block for a more practical final method.



(a) (b)

Figura 5.4: Recovered secret image: (a) shows the original VC method (50% contrast loss, secure for one transaction) and (b) shows our first proposal (75% contrast loss, secure for multiple transaction).

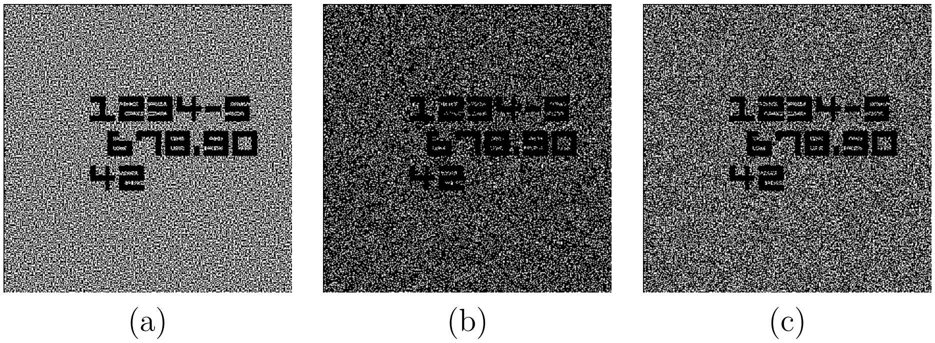
5.5.2 Decreasing non-information randomness

The method presented in the previous section effectively disables pairwise comparison as an efficient tool in the short term reconstruction of first share by a malicious observer. However, the quality of the recovered secret becomes significantly reduced, which poses a challenge for practical deployment in real-world e-banking applications in which customers have to be able to identify the information revealed by the overlay of shares in order to authenticate and authorize transactions.

This noticeable loss of quality is emphasized by the fact that non-information superpixels in the recovered image may include two, three or even four black pixels (while in the original VC method, only two black pixels are invariably present in each recovered non-information superpixel). In fact, in average one sixth of all the recovered non-information superpixels will be completely black, just like the information superpixels will be – which makes the information hard to visualize.

In order to enhance the quality of the recovered secret, we could limit the introduced freedom in choosing superpixels for the second share just enough to prevent completely black non-information superpixels from appearing in the overlaid shares, thus regaining some of the lost contrast. Therefore, instead of allowing non-information superpixels to be chosen from the six possible patterns of 2 × 2 squares during the construction of the second share, we can exclude from the choice the complement of the corresponding pattern included in the first share. While this still leaves us with five possible choices for that superpixel, we are now able to ensure that recovered non-information superpixels will include only two or three black pixels – while information superpixels will be the only completely black ones. Figure 5.5 compares the quality of the recovered secrets with the three methods for second share generation discussed so far.

We note that, even when excluding one pattern in each non-information superpixel choice for the second share, we still disable the effective reconstruction of first share by trivial pairwise comparison of two observed second shares. However, a similar statistical attack becomes viable, in the long run, when completely black non-information superpixels are avoided: given that a sufficient number of second shares had been observed by an adversary, it could be assumed that each superpixel is used as a non-information one much more often than otherwise (for instance, in our tests the amount of information superpixels account for no more than 5% of the secret image). Therefore, by verifying which were the chosen patterns for a particular position in the long term, the adversary would be able to notice that one of the six possible patterns appears much less often than the other five – thus concluding that this less-likely pattern is the complement of the correspondent in the first share. This knowledge alone is enough to guess the corresponding first share superpixel, which makes this type of statistical analysis an effective way for the adversary to succeed in reconstructing most of the first share, after observing a significant number



(a) (b) (c)

Figura 5.5: Recovered secret image: (a) shows the original VC method (50% contrast loss, secure for one transaction); (b) shows the proposal introduced in Section 5.5.1 (includes completely black non-information superpixels, secure for more-than-one transaction); and (c) shows the improved contrast proposal (no completely black non-information superpixels allowed, secure for more than one transaction).

of transactions.

By now, it should have become clear that the goal of a transaction authentication VC method is to optimally balance security aspects (i.e., the number of transactions in which the same first share can be used before enough information is available to the adversary – in the form of collected second shares – for him to be able to significantly reconstruct the first share) and quality of the recovered image (i.e., how hard it is for $User$ to distinguish the information among the non-information superpixels).

In order to achieve that balance in practice, we proceed to address the issue of choosing superpixels for second share construction with the following objectives in mind: First, we intend to increase the set of possible patterns from which superpixels can be chosen – so that statistical analysis requires a larger number of transactions to be observed by an adversary before he becomes able to reconstruct a significant portion of the secret first share; an approach for accomplishing this is suggested in Section 5.5.3. Second, we intend to make the contrast difference between information and non-information superpixels (i.e., the number of black pixels composing each type of recovered superpixel, types being information and non-information) as large as possible; this aspect of VC authentication is approached in Section 5.5.4.

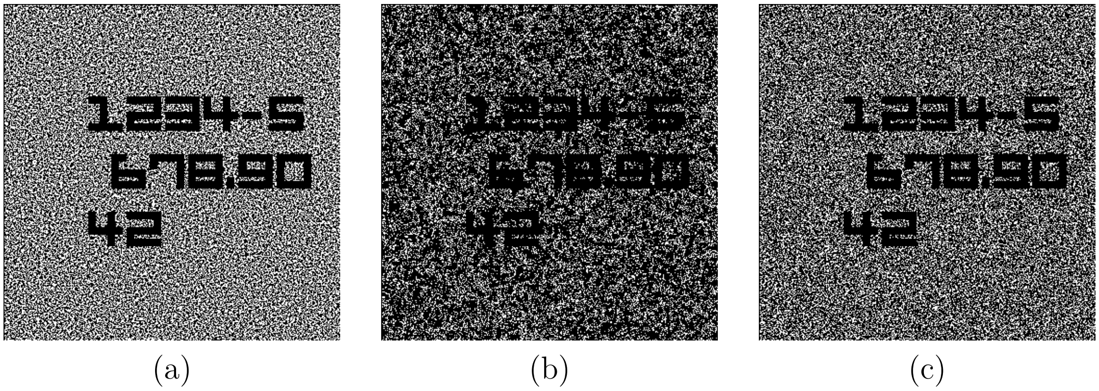
5.5.3 Larger extension factors

In Section 5.4.2 we described how, in the original VC method, the original secret image has to be extended by a factor of 2 at a pixel level – that is, how each pixel in the original image has to be expanded into a 2×2 superpixel in order to allow for its decomposition into a pair of random-looking shares. This gives us a set of six possible superpixel patterns formed by two black pixels and two white ones, from which all the superpixels to be included in both first and second shares have to be chosen (see Figure 5.1).

If, however, we use a larger-than-2 extension factor instead, we are then able to increase our set of possible superpixel patterns combinatorially. For instance, by extending each original pixel into a 4×4 superpixel, we are able to choose from 12870 possible superpixel patterns, as opposed to just six – supposing that each superpixel is comprised of the same amount of black and white pixels. Equation 5.1 shows the correlation between the extension factor f and the number of possible patterns p.

$$p = \binom{f^2}{f^2/2} \tag{5.1}$$

With a larger set of patterns to choose superpixels from, we then revisit the methods proposed in Sections 5.5.1 and 5.5.2. Figure 5.6 shows the same comparison as the one illustrated in Figure 5.5, except for the fact that we now expand each pixel from the original secret into a 4×4 superpixel before generating the pair of shares for recovery.



(a) (b) (c)

Figura 5.6: Recovered secret image with extension factor 4: (a) shows the equivalent to the original VC method (50% contrast loss, secure for one transaction); and (b) shows the equivalent to the proposal introduced in Section 5.5.1 (includes completely black non-information superpixels, secure for more-than-one transaction); and (c) shows the equivalent to the improved contrast proposal presented in Section 5.5.2 (no completely black non-information superpixels allowed, secure for multiple transactions).

We note that, by using extension factor 4 in the secret expansion, we do lose resolution[5] in the recovered secret – but we also strengthen the method proposed in Section 5.5.1

[5] This explains why the characters are larger in Figure 5.6 than in Figures 5.5 and 5.4; since the extended secret has to be the same size as the first share, we need the original secret image – in order to allow extension by a factor of 4 instead of 2 – to be one fourth of the size of the corresponding image in previously discussed approaches.

against statistical analysis of observed second shares. This happens because the set of patterns from which we can choose superpixels for each position of the second share is considerably larger than in the case where an extension factor 2 is used. This increases the number of observations required for the adversary in order to become able to recover information about the superpixels included in the first share and, therefore, makes reconstruction attacks harder to perform.

5.5.4 Thresholds: decreasing superpixel randomness for larger factors

In Section 5.5.3 we introduced larger extension factors as a method for increasing the set of possible superpixels that can be chosen in the construction of shares. This enabled us to increase the random aspect of second shares generated from a given first share, thus increasing the difficulty of reconstructing this first share by statistical analysis – which by extension allows us to reuse it for the authentication of a larger number of transactions.

Using larger superpixels can also help us to increase contrast in the recovered image. Since the contrast (and hence the ease of visualization) is given by the difference between the number of black pixels composing information and non-information superpixels in the recovered secret, having more pixels per superpixel allows us to establish a larger gap between these two types – similarly to the method described in Section 5.5.2 for 2 × 2 superpixels. For instance, if we consider an extension factor 4, shares' superpixels will be formed by eight white and eight black pixels, resulting in recovered superpixels containing any number of black pixels between eight and sixteen.

Since we have such a large set of pattern choices, we can restrict the possibilities for recovered non-information superpixels for the sake of contrast – by allowing these superpixels to contain, say, only a maximum of ten black pixels (that is, by establishing a *non-information threshold* of 10). The resulting non-information superpixels present in the recovered image would then be limited to contain between eight and ten black pixels, while information superpixels would contain sixteen black pixels. Figure 5.7 illustrates a few examples of superpixel patterns that can be found in the shares and recovered secret, in this example.

Extending the second share generation method with threshold options can be done in the following fashion: for each non-information superpixel in the first share, the corresponding to be included in the second share has to be chosen so that the superposition of both superpixels does not contain more black pixels than the threshold value. One trivial way (though unpractical due to its high computational cost) to perform that would be to randomly select a valid superpixel from the set of patterns, calculate its superposition with the pattern present in the first share and count the number of black pixels in the

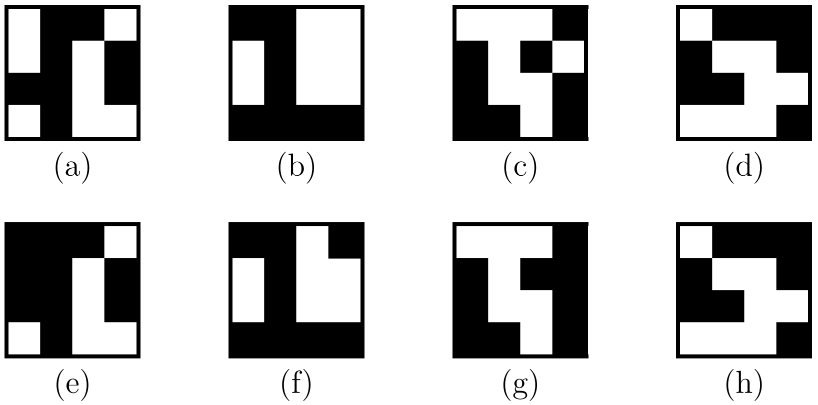


Figura 5.7: Valid extension 4 superpixels for a non-information threshold of 10. (a)–(d) illustrate superpixels that could be included in shares; (e)–(h) illustrate non-information superpixels that could be recovered upon overlaying both shares.

resulting merged pattern. If this number is less or equal than the threshold, the chosen pattern is included in the second share; otherwise, it is discarded and the process is repeated.

A considerably more efficient approach for implementing thresholds is illustrated in Figure 5.8; it is meant to be iteratively repeated for each non-information superpixel to be included in the second share. First, the method creates a local copy of the corresponding superpixel from the first share and randomly chooses the number of black pixels that the recovered superpixel will contain upon overlay (say, ten, in the illustrated example); this number is limited by the interval between the number of black pixels already contained in the local copy and the threshold value. The method then randomly chooses enough white pixels in the copy and paint them black, producing a derived superpixel that contains the chosen amount of black pixels. This process is summarized in Step (a), and outputs the pattern that will be recovered when that first share superpixel is overlaid with the second share superpixel we are constructing.

In Step (b), the produced recovered superpixel is XORed with the corresponding first share superpixel, thus resulting in a partial second share superpixel that, for now, contains only the black pixels painted in Step (a). This superpixel is completed in Step (c) by randomly choosing pixels that are black both in the recovered superpixel outputed in Step (a) and in the corresponding first share superpixel, and painting them black in the partial second share superpixel – until it contains enough black pixels (in this example, eight) to be considered a valid superpixel. The output of Step (c) is the final superpixel that will then be included in the same position in the second share as the first share superpixel taken as input in Step (a); upon overlaying both shares during secret recovery, the non-information superpixel output in Step (a) – ensured by design to be bounded by the established threshold – will be revealed.

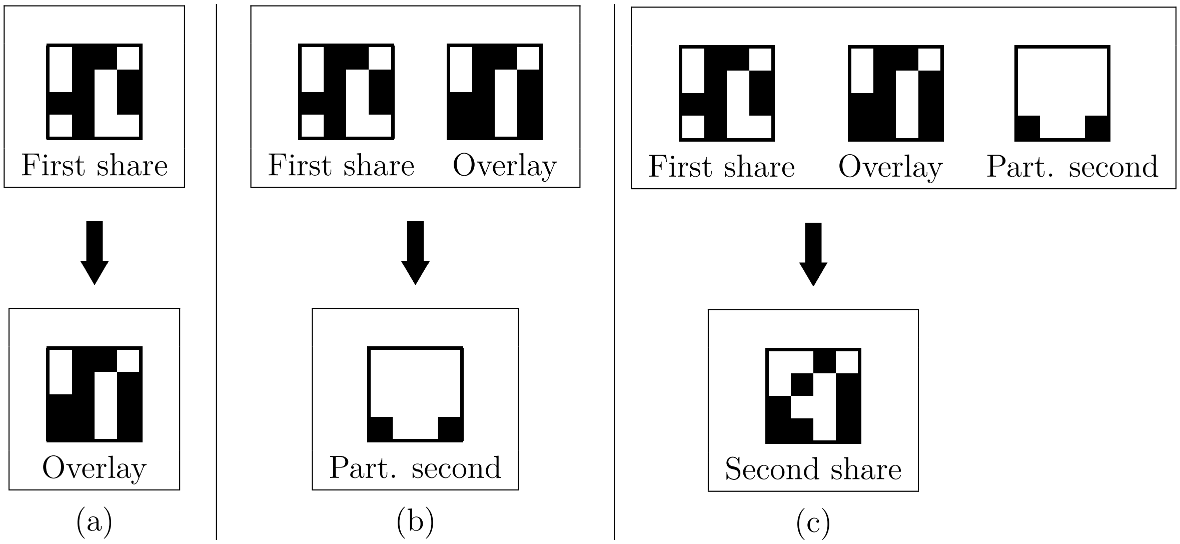
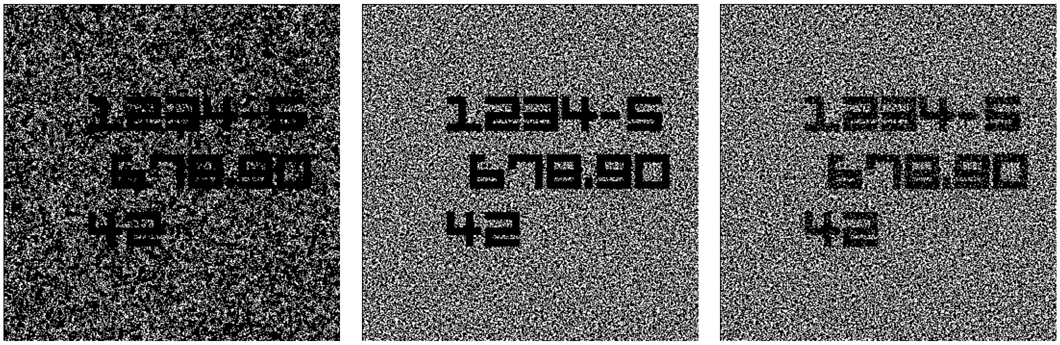


Figura 5.8: Efficient superpixel selection algorithm for second share construction with threshold

Information threshold

Naturally, the concept of threshold can be extended also for information superpixels. In this case, however, the threshold determines a minimum limit for the number of black pixels to form a valid recovered information superpixel, as opposed to a maximum limit for recovered non-information superpixels. This means that we do not have to constrain ourselves to information superpixels containing no less than sixteen black pixels (for extension factor 4), which would only provide us with one possible complement choice for each information superpixel; if for instance we establish an information threshold of 14, we allow recovered information superpixels to contain either fourteen, fifteen or sixteen black pixels, thus introducing extra contrast degradation, but increasing the number of possible choices for information regions to the same extent as we have done for non-information regions.

When larger extension factors are used, the contrast gain of threshold-based selection is noticeable, if compared to the use of the random selection method presented in Section 5.5.3. Figure 5.9 illustrates examples of second shares produced with random, one-threshold and two-threshold superpixel selection methods for comparison.



(a) (b) (c)

Figura 5.9: Recovered secret image with extension factor 4: (a) shows the result when the method presented in Section 5.5.3 is applied (no threshold); (b) shows the threshold-based selection method (with non-information threshold of 10); and (c) shows the double-threshold-based selection method (with non-information threshold of 10 and information threshold of 14).

5.5.5 Isolating information from randomly-chosen non-information superpixels

In Section 5.5.4 we presented information and non-information thresholds for respectively expanding and reducing the set of possible choices for each superpixel to be included in the second share. While this approach makes it harder for the adversary to guess information superpixels (since several possible patterns can now match each superpixel in the first share, in order to result in a recovered information superpixel – as opposed to just one pattern, when information thresholds are not used), the opposite occurs for non-information superpixels, in comparison to the truly-random, threshold-free alternative presented in Section 5.5.3. Nevertheless, as Figure 5.9 illustrates, usability (i.e., ease of visualization of the recovered information) is significantly enhanced when thresholds are used, due to the contrast gain introduced by increasing the difference between the number of black pixels contained in recovered information and non-information superpixels.

In order to take advantage of this contrast gain, while at the same time approaching the level of robustness against non-information superpixel guessing achieved by the method described in Section 5.5.3, we can restrict the regions in which the non-information threshold is applied – specifically, to the non-information superpixels belonging to an established region around the included secret information. The main idea behind this approach is the fact that the contrast between information and non-information superpixels does not necessarily need to be enhanced relatively to the whole recovered image – it is enough, for ease of visualization purposes, to limit the increase contrast between information neighboring non-information superpixels instead. This allows for a loss of randomness only in a small percentage of the non-information superpixels included in the second share, rather than in the majority of truly-random non-information superpixels chosen to compose that share. Since the relative position of the information superpixels in the image (and consequently, the position of the thresholded non-information superpixels around it) is randomly chosen at run time by *Bank* for each transaction, the adversary cannot accurately determine where to perform the guessing attack with a limited amount of observed second shares.

We note that this method can be applied not only when larger extension factors are used, but also for the extension factor 2 used in the original VC method. Figure 5.10 shows four examples of recovered secrets with limited-region non-information threshold.

5.6 A robust VC method for e-banking authentication with reusable first share

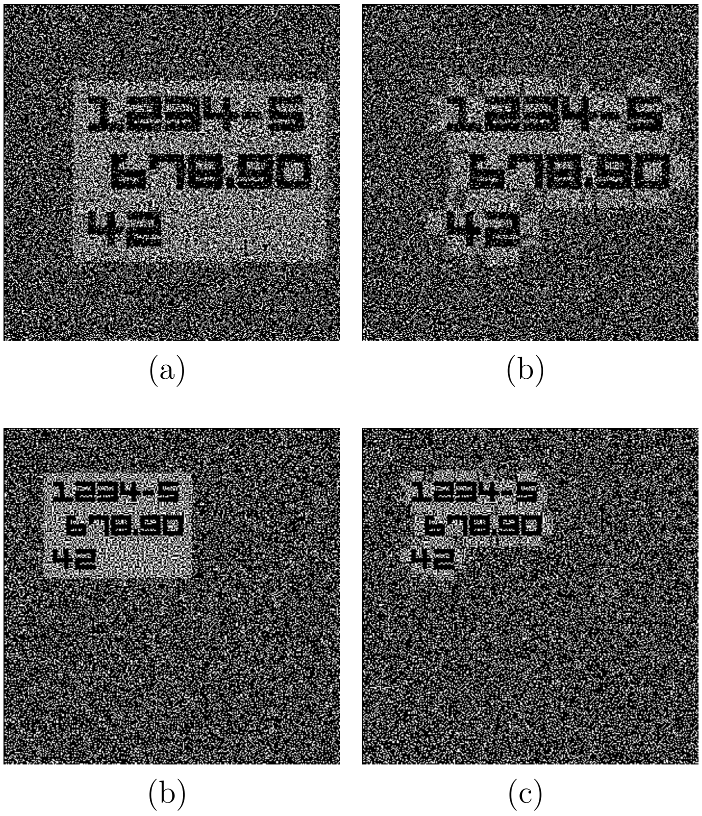
In Section 5.4 we introduced our general concept of reusable VC for e-banking authentication, which we further extended in Section 5.5 by proposing and analyzing several approaches for both making first share reconstruction attacks harder and facilitating visual recovery of the secret information by $User$. In this section we present our finished method for second share generation, which consists of a combination of the approaches presented in Section 5.5. We also present brief discussions on security, usability and logistical aspects of the proposed solution taking realistic $User$ and $Bank$ requirements into account. We note that the hereby proposed method refers to the method $M3$, introduced in Section 5.4.3, which is a component of the protocol illustrated in Figure 5.3.

Our practical version of reusable VC is based on the method described in Section 5.5.5, with a fixed extension factor 4; since we foresee a scenario in which $User$ VC cards should be small enough to be stored in the customer's wallet – such as a standard credit card sized transparency, for instance – larger extension factors would cause the usable physical area available for embedding information to be significantly reduced, thus increasing the probability of information intersections between multiple second shares[6].

In order to strengthen the method against reconstruction attacks, the following parameters should be randomly set in run time by $Bank$ for each transaction:

1. **Transaction authentication number (TAN)**: A random number generated in Step 3 of the authentication protocol, which shall be recovered by $User$ and resent

[6] The extension factor can vary from customer to customer, if *Bank* so desires. Variable extension factors can be used to provide different levels of security, usability and first share reusability for different categories of account holders.



(a) (b)

(b) (c)

Figura 5.10: Recovered secret image with non-information threshold limited to the neighbor region around the information: (a) shows a shape-fixed rectangular region (extension factor 4, non-information threshold of 10 and information threshold of 14); (b) shows an irregular region (extension factor 4, non-information threshold of 10 and information threshold of 14); (c) shows a shape-fixed rectangular region (extension factor 2,non-information threshold of 2); and (d) shows an irregular region (extension factor 2, non-information threshold of 2).

to *Bank* in order to authorize the transaction to be committed.

2. **Position of the transaction information**: The coordinates where TAN and $TInfo$ will be inserted in the secret image $SecIm$.

3. **Non-information and information thresholds**: By allowing variable thresholds, the method makes reconstruction attacks harder – since the number of selectable superpixels per superpixel contained in the first share varies from one transaction to the other. We described the use of thresholds in Section 5.5.4.

4. **Shape and size of the neighbor region for non-information threshold**: Non-information thresholds should be applied to a limited region around the information

superpixels, as described in Section 5.5.5, but we believe fixed-shaped regions to be easier to be detected in reconstruction attacks. By allowing region shape and size to be variable, we put yet another obstacle for the adversary's statistical recovery of the $User$'s secret share.

5. **Text modifiers**: The method should also provide a set of suitable font shapes and font sizes for selection in run-time, as well as different rotation angles and text distortions during secret image generation. This provides the method with CAPTCHA-like features, thus making it harder for the adversary to predict information placement during authentication iterations [5].

All these variable parameters can be chosen randomly by *Bank* from a predefined set of options regardless of $User$ knowing which choices were made. In other words, the process performed by $User$ in order to recover the information and authenticate the transaction is the same and independent of which parameters were selected, as is described by Steps 7-10 of Figure 5.3. Therefore, the *Bank*-side implementation of method $M3$ should take usability aspects into account when defining each set of selectable parameters – for instance, a sufficient gap should be kept between non-information and information thresholds to ensure ease of secret recovery, and the set of selectable font shapes and sizes should include only options that are easy to visualize upon overlay. We also note that these sets can be freely altered by *Bank* as required, without the need of reissuing any $User$ VC card – which presents itself as a logistical advantage over other methods.

In order to be implemented effectively, our method requires that each $User$'s VC card be generated by taking that user's smartphone's technical specifications into account. In that context, display size and pixel density play an important role. Our proof-of-concept tests were performed on Apple's iPhone 4, which currently relies on the Retina Display technology capable of providing a 326 ppi pixel density. We note that our method is capable of working with any smartphone brand or model, being limited only by the pixel density capabilities of the high-precision printer required to manufacture the VC cards.

As for usability, we note that overlaying the VC card with the second share can be a hard task, since the card has to be positioned with pixel-level precision on the device display in order to allow the information to be recovered. This task can be made easier by providing each $User$ with a physical attachable holder specifically designed for his smartphone model. Such approach is not only low-cost in comparison to similar e-banking solutions, but also reduces $User$'s task to authenticate a transaction to as much as inserting his VC card into the holder, and attaching the holder to the smartphone when required.

Capítulo 6

Conclusion

In this thesis, we approached the design of cryptographic solutions and how they, by not taking into account users' particular requirements and related human factors, not always succeed in providing the guarantees they were meant for. We approached this subject by analyzing two cases of study, namely the electronic commerce of digital items and Internet banking authentication. In this chapter, we conclude our discussion by revisiting some of the topics so far presented.

By studying the currently adopted model for buying and selling digital content from a fair exchange perspective (Chapter 2), we observed several examples of unfair outcomes in real-world transactions. We believe that such problems arise from the fact that the present applications in this context fail to consider specific subtleties of the digital content being offered – such as indescribability – and hence concern the item validation problem of fair exchange. As such, we firmly believe that further research aimed at enhancing consumer satisfaction and trust in the e-commerce business model should focus on the aspects presented in this work. Neither currently published fair exchange protocols, nor the currently implemented e-commerce business models seem able to provide any level of fairness – by means of enabling unambiguous dispute processes for handling exceptions – to both customers and sellers simultaneously, given the characteristics of current digital products. Specifically in the context of digital multimedia selling, which is increasingly becoming the mainstream form of media consumption among the general public, new solutions should be proposed to reduce customer losses and increase reliance on e-commerce transactions.

We stress that, to the best of our knowledge, the problem of item validation of fair exchange protocols has barely been studied [19, 63]. The item validation step is often briefly described in protocol specifications, if not completely overlooked. By treating validation techniques as nothing but a byproduct of protocol design, researchers have neglected to approach a hard problem in protocol design – one that might lead to fruitful discussions and further enhancements on the state of the art of diplomatic protocol design.

The fair exchange of indescribable items, for instance, has only been first approached recently [19], and techniques suitable for the optimistic fair exchange of those items are yet to be proposed. The reversible degradation concept is the first to address the exchange of indescribable items optimistically (i.e., by providing unambiguous dispute resolution through the use of atomic validation artifacts). Furthermore, other alternative models for multimedia purchase, presented in Section 3.5, do not seem suitable for consistently addressing the hereby discussed issues.

We presented a SECC-based technique that, to the extent of our knowledge, is the first implementable instantiation of the concept of reversible degradation [63] (Chapter 3). Our implementation addresses the validation of frame-structured multimedia items consisting of perceptual information, and is described in two methods: a degradation method, which allows an arbitrary amount of degradation to be added to the item and which outputs not only a degraded version of the item, but also a restoring key for reversing the process; and a recovery method, which allows the full recovery of the original item from its degraded copy, provided that the restoring key is known. By using SECCs as the basis for our instantiation, we therefore rely both on the hardness of the problem of correcting random errors in digital information without redundancy and on atomic validation artifacts, aimed at enabling unambiguous dispute resolution, to justify our security claims.

As a final remark on our reversible degradation implementation, even though we provided experimental data regarding the proposed technique, we stress that optimal parameter values for the degradation of items (which depend not only on the nature of the items itself, but also on each *Seller*'s perspective on quality of service and customer-provider relationship) should be taken into account by specific applications of the method. Specifically, for practical scenario deployment, it should be evaluated what would be a good level of degradation for the intended items – which affects both security (how hard it is for a malicious *Buyer* to restore the original item from the degraded version without knowledge of the key) and quality of service (how hard it is for an honest *Buyer* to validate the degraded version). Also, the restoring key size is affected by both code parameters and desired degradation level, and should be minimized in order to achieve optimal data transmission efficiency. All these issues remain the focus of our future work.

As for our experience with item-aware protocol design (Chapter 4), we firmly believe that the current generic item approach to fair exchange introduces more problems to fair exchange solutions than it solves – a misleading oversimplification of a rather delicate, context-sensitive process. We analyzed several inherent aspects of digital items and presented an interaction-oriented discussion on how the designer can take advantage of item characteristics, in order to simplify and improve the accuracy of such protocols by designing them in an item-aware fashion.

By taking into account our remarks on the interactions between properties (Section 4.2), protocol designers may avoid disruptive effects that might undermine protocol goals when the items to be exchanged hold particular characteristics. These remarks address not only security aspects, but also quality of service and efficiency aspects (as well as human requirements), which might be of vital importance for real-world systems.

We illustrate the benefits of this particular discussion with an example of how protocol design can be made significantly more accurate by acknowledging such inherent aspects of items. The protocol produced through our item-aware design example includes mechanisms that provide more-accurate item validation and easier dispute resolution – thus providing robustness against specific, context-related issues posed by real-world scenarios, such as “buying a pig in a poke” and no-return policies.

We finish our discussion on item-aware protocol design by stating that we are currently unaware of any previous works concerning fair exchange that provide alternative models to the traditional generic protocol design and, as such, further research should be conducted on this subject. By further identifying interesting item properties and proposing specific item-related dispute processes and item validation techniques (i.e., the reversible degradation method), fair exchange researchers should be able to address several issues that are taken lightly in the current model. Therefore, we believe that abandoning the generic item-oriented model of fair exchange is, at least in the context of real-world applications, essential for future proposals intended as suitable solutions for practical scenarios [63, 61].

Finally, by taking into account current realistic aspects of security and human factors, we presented a robust authentication method for sensitive transaction scenarios that does not rely on any assumptions that might be too hard to guarantee (Chapter 5). We considered our problem scenario under an up-to-date adversarial model (i.e., the modern adversary model), which reflects the state-of-the-art in malicious capabilities, as well as both server-side logistical concerns and client-side typical behavior and limitations.

By looking beyond technical aspects, we were able to design a solution that does not rely on the possibly compromised device’s computational capabilities and that restricts the impact caused by human misconceptions about safe behavior. Specifically, our approach to designing cryptographic solutions takes into account the fact that, even though several traditional Cryptography artifacts are many times provably secure (from a theoretical perspective), they are not for realistic scenarios – where human factors can also undermine Security requirements – and therefore cannot be seen as guaranteed solutions.

As for the human factors of Security solutions, we believe that such approaches as Visual Cryptography and reversible degradation hold an interesting aspect for future topics of Security research: they allow for concepts that might be otherwise hard to grasp for the average user to be intuitively assimilated and instinctively trusted. In other words, previous approaches to Security method design have always attempted to protect the user's needs without his active intervention, thus being effectively invisible. However, as far as the perception of Security goes, there might be other ways to design secure cryptographic methods without excluding human factors from their critical path; instead, those methods can take advantage of those same factors in order to more-accurately achieve their goals. We have illustrated this concept by proposing two solutions for two very distinct scenarios: in one case, we relied on the user's abilities to recognize multimedia content with his own senses, so that he can decide whether to buy it or not; in another case, we delivered to the user the task to completely decrypt sensitive information with nothing else than his own visual capabilities.

Finally, we believe the design of Security methods that not only do not disregard, but also take advantage of human factors, to be a promising topic for future research in Cryptography. While it is obvious that Security solutions should always ensure information protection, we should not forget that, in general terms, they are ultimately designed to be used by humans; to disregard the way most humans interact with those solutions – and the very way they perceive, behave towards and understand Security itself – can be and usually is, in many cases, just as good as treating a patient's condition with the wrong medicine.

Referências Bibliográficas

[1] Supplementary test data. `http://www.lca.ic.unicamp.br/~fpiva/testdata.zip`. Online; accessed 25-February-2014.

[2] The Invisible Web Unmasked: TrendLabsSM 3Q 2013 Security Roundup. Technical report, 2013.

[3] Martín Abadi and Roger Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, 1996.

[4] Anne Adams and Martina Angela Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, December 1999.

[5] Luis Ahn, Manuel Blum, NicholasJ. Hopper, and John Langford. CAPTCHA: Using Hard AI Problems for Security. *Advances in Cryptology — EUROCRYPT 2003*, 2656:294–311, 2003.

[6] A Alaraj and M Munro. An e-commerce fair exchange protocol for exchanging digital products and payments. *International Conference on Digital Information Management*, 1:248–253, 2007.

[7] Fadi Aloul, Syed Zahidi, and Wasim El-Hajj. Multi Factor Authentication Using Mobile Phones. *International Journal of Mathematics and Computer Science*, 4(2):65–80, 2009.

[8] Amazon Legal Department. Product Return Policies: Digital Content. Online; accessed 25-February-2014.

[9] Amazon Legal Department. Amazon MP3 Music Service: Terms of Use, 2005.

[10] A Asokan. *Fairness in Electronic Commerce.* PhD thesis, University of Waterloo, 1998.

[11] N. Asokan, P.A. Janson, M. Steiner, and M. Waidner. The state of the art in electronic payment systems. *Computer*, 30(9):28–35, 1997.

[12] N Asokan, Victor Shoup, and Michael Waidner. Optimistic Fair Exchange of Digital Signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):593–610, 1999.

[13] Giuseppe Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, pages 138–146, New York, NY, USA, 1999. ACM.

[14] Giuseppe Ateniese. Verifiable encryption of digital signatures and applications. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):1–20, 2004.

[15] Gildas Avoine and Serge Vaudenay. Optimistic Fair Exchange Based on Publicly Verifiable Secret Sharing. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, volume 3108 of *Lecture Notes in Computer Science*, pages 74–85. Springer Berlin Heidelberg, 2004.

[16] Gildas Avoine and Serge Vaudenay. Optimistic fair exchange based on publicly verifiable secret sharing. In *Information Security and Privacy: 9th Australasian Conference, ACISP 2004*, volume 3108 of *Lecture Notes in Computer Science*, pages 74–85, 2004.

[17] Abhilasha Bhargav-Spantzel, Anna C. Squicciarini, Shimon Modi, Matthew Young, Elisa Bertino, and Stephen J. Elliott. Privacy preserving multi-factor authentication with biometrics. *Journal of Computer Security*, 15(5):529–560, 2007.

[18] R W H Bons, R M Lee, and R W Wagenaar. *Obstacles for the Development of Open Electronic Commerce.* Erasmus University, Erasmus University Research Institute for Decision and Information Systems (EURIDIS), 1995.

[19] A Bottoni, G Dini, and T Stabell-Kulø. A methodology for verification of digital items in fair exchange protocols with active trustee. *Electron Commerce Res*, 7(2):143–164, 2007.

[20] Colin Boyd and Anish Mathuria. *Protocols for authentication and key establishment.* Springer, 2003.

[21] F. Callegati, W. Cerroni, and M. Ramilli. Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security & Privacy*, 7(1):71–81, 2009.

[22] C. Castelfranchi. The role of trust and deception in virtual societies. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, page 8. IEEE Comput. Soc, 2001.

[23] Antitza Dantcheva, Carmelo Velardo, Angela D'Angelo, and Jean-Luc Dugelay. Bag of soft biometrics for person identification. *Multimedia Tools and Applications*, 51(2):739–777, October 2010.

[24] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, March 1983.

[25] W Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. Security automation considered harmful? In *Proceedings of the 2007 Workshop on New Security Paradigms - NSPW '07*, page 33, New York, New York, USA, July 2008. ACM Press.

[26] Xing Fang and Justin Zhan. Online Banking Authentication Using Mobile Phones. In *5th International Conference on Future Information Technology*, pages 1–5. IEEE Computer Society, 2010.

[27] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '11*, page 3, New York, New York, USA, October 2011. ACM Press.

[28] Jen-Bang Feng, Hsien-Chu Wu, Chwei-Shyong Tsai, Ya-Fen Chang, and Yen-Ping Chu. Visual secret sharing for multiple secrets. *Pattern Recognition*, 41(12):3572–3581, December 2008.

[29] Juan A. Garay, Markus Jakobsson, and Philip D. MacKenzie. Abuse-Free Optimistic Contract Signing. In *Advances in Cryptology - Crypto '99*, pages 449–466. Springer Berlin Heidelberg, August 1999.

[30] Felix C. Gartner, Henning Pagnia, and Holger Vogt. Approaching a Formal Definition of Fairness in Electronic Commerce. In *18th IEEE Symposium on Reliable Distributed Systems*, pages 354–359. IEEE Computer Society, 1999.

[31] V. Goppa. A New Class of Linear Correcting Codes. *Problems of Information Transmission*, 6(3):207–212, 1970.

[32] Han-Na You, Jae-Sik Lee, Jung-Jae Kim, and Moon-Seog Jun. A study on the two-channel authentication method which provides two-way authentication in the Internet banking environment. In *5th International Conference on Computer Sciences and Convergence Information Technology*, pages 539–543. IEEE, November 2010.

[33] Petr Hanaek, Kamil Malinka, and Jiri Schafer. E-banking security - comparative study. In *42nd Annual IEEE International Carnahan Conference on Security Technology*, pages 326–330. IEEE Computer Society, October 2008.

[34] Shane Hathaway. Reed-Solomon Python Extension Module, 2005.

[35] A. Hiltgen, T. Kramp, and T. Weigold. Secure Internet banking authentication. *IEEE Security & Privacy Magazine*, 4(2):21–29, March 2006.

[36] P G W Keen. *Electronic commerce relationships: Trust by design.* Prentice Hall PTR, 2000.

[37] A Kiayias and M Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 54(6):2752–2769, January 2008.

[38] S Kremer, O Markowitch, and J Zhou. An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17):1606–1621, November 2002.

[39] S. Kwong. An algorithm for removable visible watermarking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(1):129–133, January 2006.

[40] Gerard Lacoste, Birgit Pfitzmann, Michael Steiner, and M Waidner. SEMPER - Secure Electronic Marketplace for Europe. *Lecture Notes in Computer Science (LNCS)*, 1854, August 2000.

[41] Young Sil Lee, Nack Hyun Kim, Hyotaek Lim, Heungkuk Jo, and Hoon Jae Lee. Online banking authentication system using mobile-OTP with QR-code. *5th International Conference on Computer Sciences and Convergence Information Technology*, pages 644–648, November 2010.

[42] Shiguo Lian. *Multimedia Content Encryption: Techniques and Applications.* CRC Press, 2009.

[43] Panagiotis Louridas. Some guidelines for non-repudiation protocols. *SIGCOMM Comput. Commun. Rev.*, 30(5):29–38, 2000.

[44] Mikko Loytynoja, Nedeljko Cvejic, and Tapio Seppanen. Audio protection with removable watermarking. In *2007 6th International Conference on Information, Communications & Signal Processing*, pages 1–4. IEEE, 2007.

[45] Olivier Markowitch and Steve Kremer. An Optimistic Non-repudiation Protocol with Transparent Trusted Third Party. pages 363–378, October 2001.

[46] Olivier Markowitch and Yves Roggeman. Probabilistic Non-Repudiation without Trusted Third Party. In *Second Workshop on Security in Communication Network*, 1999.

[47] Olivier Markowitch and Shahrokh Saeednia. Optimistic Fair Exchange with Transparent Signature Recovery. In *Financial Cryptography*, volume 2339, pages 339–350. Springer Berlin Heidelberg, February 2002.

[48] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, 1978.

[49] Silvio Micali. Simple and fast optimistic protocols for fair electronic exchange. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 12–19, New York, NY, USA, 2003. ACM Press.

[50] L Minder. Cryptography based on error correcting codes. *algo.epfl.ch*, 2007.

[51] Moni Naor and Benny Pinkas. Visual authentication and identification. In Burton Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 322–336. Springer Berlin / Heidelberg, 1997.

[52] Moni Naor and Adi Shamir. Visual Cryptography. In *Eurocrypt 94*, pages 1–12, 1994.

[53] A Nenadic, Ning Zhang, and Stephen Barton. FIDES - a Middleware ECommerce Security Solution. In *The 3rd European Conference on Information Warfare and Security (ECIW*, pages 295–304, 2004.

[54] Aleksandra Nenadic, Ning Zhang, Qi Shi, and Carole Goble. DSA-Based Verifiable and Recoverable Encryption of Signatures and Its Application in Certified E-Goods Delivery. In *IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service*, pages 94–99. IEEE Computer Society, 2005.

[55] Donal O'Mahony, Hitesh Tewari, and Michael Peirce. *Electronic Payment Systems.* Artech House, June 1997.

[56] Rolf Oppliger, Ruedi Rytz, and Thomas Holderegger. Internet Banking: Client Side Attacks and Protection Mechanisms. *Computer*, 42(6):27–33, 2009.

[57] H Pagnia and F C Gartner. On the impossibility of fair exchange without a trusted third party. *Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany Technical Report TUD-BS-1999-02*, 1999.

[58] H Pagnia, H Vogt, and F Gartner. Fair exchange. *The Computer Journal*, 46(1):55–75, January 2003.

[59] Magdalena Payeras-Capellà, Josep Lluís Ferrer-Gomila, and Llorenc Huguet-Rotger. Achieving fairness and timeliness in a previous electronic contract signing protocol. In *ARES*, pages 717–722. IEEE Computer Society, 2006.

[60] F R Piva, J R M Monteiro, and R Dahab. Regarding timeliness in the context of fair exchange. In *Network and Service Security, 2009. N2S '09. International Conference on*, pages 1–6, June 2009.

[61] Fabio Piva and Ricardo Dahab. Modern fair exchange protocol design: Dealing with complex digital items. In *XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, Manaus/Brazil, 2013.

[62] Fabio R Piva. Verificação formal de protocolos de trocas justas utilizando o método de espaços de fitas. Master's thesis, UNICAMP, 2009.

[63] Fabio R Piva and Ricardo Dahab. E-commerce and fair exchange: The problem of item validation. In *International Conference on Security and Cryptography (SECRYPT)*, pages 317–324, Seville, Spain, 2011. SciTePress Digital Library.

[64] Fabio R Piva, José R M Monteiro, and Ricardo Dahab. Strand spaces and fair exchange: More on how to trace attacks and security problems. In *VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, September 2007.

[65] Fabio R Piva, José R M Monteiro, Augusto J Devegili, and Ricardo Dahab. Applying Strand Spaces to Certified Delivery Proofs. In *VI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, September 2006.

[66] Havard Raddum, LarsHopland Nestas, and Kjell Jorgen Hole. Security Analysis of Mobile Phones Used as OTP Generators. In *Information Security Theory and Practices: Security and Privacy of Pervasive Systems and Smart Devices*, volume 6033 of *Lecture Notes in Computer Science*, pages 324–331. Springer Berlin Heidelberg, 2010.

[67] I Ray. Fair exchange in e-commerce. *ACM SIGecom Exchanges*, 3(2):9–17, 2002.

[68] Indrakshi Ray and Indrajit Ray. An optimistic fair exchange e-commerce protocol with automated dispute resolution. In *EC-Web*, pages 84–93, 2000.

[69] Indrakshi Ray and Indrajit Ray. An anonymous fair exchange e-commerce protocol. In *International Workshop on Internet Computing and Ecommerce*, pages 172–179. IEEE Computer Society, 2001.

[70] I S Reed and G Solomon. Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, March 1960.

[71] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy van der Horst, and Kent Seamons. Confused Johnny: when automatic encryption leads to confusion and mistakes. In *Proceedings of the Ninth Symposium on Usable Privacy and Security - SOUPS '13*, page 1, New York, New York, USA, July 2013. ACM Press.

[72] S Shlien. Guide to MPEG-1 audio standard. *IEEE Transactions on Broadcasting*, 40(4):206–218, January 1994.

[73] D K Smetters and R E Grinter. Moving from the design of usable security technologies to the design of useful secure applications. In *Proceedings of the 2002 workshop on New security paradigms - NSPW '02*, page 82, New York, New York, USA, September 2002. ACM Press.

[74] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. An erasures-and-errors decoding algorithm for Goppa codes. *IEEE Transactions on Information Theory*, 22(2):238–241, March 1976.

[75] Yanbin Sun, Lize Gu, Sihan Qing, Shihui Zheng, Yixian Yang, and Yan Sun. New Optimistic Fair Exchange Protocol Based on Short Signature. In *Second International Conference on Communication Software and Networks (ICCSN)*, pages 99–104, Singapore, 2010. IEEE Computer Society.

[76] Yao-Hua Tan. A Trust Matrix Model for Electronic Commerce. *Trust Management*, 2692:33–45, May 2003.

[77] Yao-Hua Tan and W. Theon. Formal aspects of a generic model of trust for electronic commerce. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, volume vol.1, page 8. IEEE Comput. Soc, 2000.

[78] Y.H. Tan and W. Thoen. A generic model of trust in electronic commerce. *International Journal of Electronic Commerce*, 5(2):61—-74, 2000.

[79] F Javier Thayer, Jonathan C Herzog, and Joshua D Guttman. Strand Spaces: Proving Security Protocols Correct. *Journal of Computer Security*, 7(2–3):191–230, 1999.

[80] Hien Thi Thu Truong, Eemil Lagerspetz, Petteri Nurmi, Adam J. Oliner, Sasu Tarkoma, N. Asokan, and Sourav Bhattacharya. The Company You Keep: Mobile Malware Infection Rates and Inexpensive Risk Indicators. *Computing Research Repository (CoRR)*, abs/1312.3, 2013.

[81] Mikko Valimaki and Ville Oksanen. DRM interoperability and intellectual property policy in europe. *European Intellectual Property Review*, 26(11):562–568, 2006.

[82] Holger Vogt. Asynchronous optimistic fair exchange based on revocable items. In *Financial Cryptography*, pages 208–222, 2003.

[83] H Wang and H Guo. Fair payment protocols for e-commerce. *IFIP Advances in Information and Communication Technology*, pages 227–245, 2004.

[84] James Wayman, Anil Jain, Davide Maltoni, and Dario Maio. An Introduction to Biometric Authentication Systems. *Biometric Systems*, pages 1–20, 2005.

[85] Thomas Y. C. Woo and Simon S. Lam. A lesson on authentication protocol design. *Operating Systems Review*, 28(3):24–37, 1994.

[86] C.C. Wu and L.H. Chen. *A Study On Visual Cryptography.* PhD thesis, Institute of Computer and Information Science, National Chiao Tung University, 1998.

[87] Jianying Zhou, Robert H. Deng, and Feng Bao. Evolution of fair non-repudiation with TTP. In *ACISP '99: Proceedings of the 4th Australasian Conference on Information Security and Privacy*, pages 258–269, London, UK, 1999. Springer-Verlag.

[88] Jianying Zhou, Robert H. Deng, and Feng Bao. Some Remarks on a Fair Exchange Protocol. In *Public Key Cryptography*, pages 46–57. Springer Berlin Heidelberg, January 2000.

[89] Min Zuo and Jianhua Li. Constructing fair-exchange p2p file market. In *Proceedings of the 4th International Conference on Grid and Cooperative Computing*, pages 941–946, 2005.

[90] Mary Ellen Zurko and Richard T Simon. User-centered security. In *Proceedings of the 1996 workshop on New security paradigms - NSPW '96*, pages 27–33, New York, New York, USA, September 1996. ACM Press.