Visual Crypto Displays Enabling Secure Communications

Pim Tuyls, Tom Kevenaar, Geert-Jan Schrijen, Toine Staring, Marten van Dijk

Philips Research Laboratories, 5656 AA Eindhoven, the Netherlands

Abstract

In this paper we describe a low-tech and user friendly solution for secure two-way communication between two parties over a network of *untrusted* devices. We present a solution in which displays play a central role. Our approach guarantees privacy and allows to check the authenticity of information presented on displays. Furthermore, we provide the user with a secure return channel. To this end we propose to provide every user with a small *decryption display* which is, for example, integrated in a credit card and requires very limited computing power. The authentication and security are based on visual cryptography which was first introduced by Naor and Shamir in 1994. We solve some practical shortcomings of traditional visual cryptography and develop protocols for two-way authentication and privacy in untrusted environments.

1 Introduction

As the world is moving into a real ambient intelligence environment, it is expected that telecommunication devices will be omnipresent and that they can be easily connected to a public network. In almost any situation and location it will be possible to check your bank account, place orders at the stock market, transmit pictures, etc. Moreover, to facilitate human interaction with these devices, most information will be presented in graphical form on a display.

In general, public networks have an open architecture and can easily be accessed. This causes a real threat to the authenticity of the information, its security and its privacy [7]. Imagine a scenario in which Alice wants to transfer money to a friend while she is on vacation. Via the access point in her hotel, she connects with her PDA to the web-site of her bank. As soon as she notices the small padlock in the corner of her Internet browser application, she starts the Internet banking session. She uses her Internet banking token (issued by the bank) for identification and for signing her money transfer order. This way of doing presents many risks. Although Alice thinks she is involved in a secure transaction with her bank (the padlock indicates a secure connection), in fact she might not be. For example, the server in the hotel might be setting up a man-in-the-middle attack and route traffic to a fake web-site that is controlled by the hotel itself [1]. If the hotel succeeds in this attack, transferred amounts and destination bank account of the transaction can easily be altered without notice to the user. A second risk is the fact that in earlier occasions Alice's PDA was connected to untrusted networks and might be contaminated with viruses, sniffer programs or Trojan horses and thus can not be trusted either [7]. Again this means that a secure connection might be compromised without notice to the user.

Clearly, there is a large variety of cryptographic techniques available [10, 14] to ensure the authenticity and protect privacy when information is transmitted over a network. However, these techniques can not be applied straightforwardly for communication over untrusted networks since it is assumed that decryption is performed on a trusted device.

In this paper, we present solutions for the above mentioned problem. To provide integrity, authentication and confidentiality of messages that are transmitted through a public untrusted channel, a user will have to carry her own (trusted) *decryption display*. This display can easily be attached to a hand-held device [1] (e.g. mobile phone, PDA) or to another non-trusted terminal (e.g. ATM, PC in Internet cafe, etc.). Its main requirements are: cheap, low weight, limited computing power and small size. The decryption display is only equipped with simple dedicated hardware such as a pseudo random number generator and the interaction with the untrusted hand-held device is purely optical. Therefore there will be no means for Trojan horses or viruses to enter the decryption display.

Security problems related to the human-computer interface have also been investigated by Matsumoto [8, 9]. Their approach is based on visual images that have to be remembered by the user and therefore require an extra effort. Another approach to visual identification and authentication is that of [4]. Their solution suits well with larger screens and requires a trustworthy camera, but no integrity or confidentiality can be guaranteed.

This paper is organised as follows. The model and the problem setting will be explained in Section 2 and Section 3 gives a detailed explanation of the way an efficient visual cryptography system can be built by making use of the physical properties of light and Liquid Crystal Displays (LCD). Furthermore, in Section 4 protocols are developed to establish a private and authenticated two-way channel. In Section 5 we demonstrate the feasibility of the concept and we conclude in Section 6.

2 The Model and Problem Setting

2.1 The Model

In this section, we present an abstract model of the situation we have in mind (Fig. 1). The model consists of the following components:

- There is a human user, Alice who has the capabilities (vision, colour recognition) of a normal human being.

[1] The trusted display can be correctly aligned on the untrusted display by providing a solid frame into which it has to be entered or by equipping it with extra sensors which automatically read position information from the untrusted terminal. This will be a topic of further research.

– There is a trusted computer TC (with normal computing power) that knows a secret key K associated with Alice [2].
– Both communicate with each other through an *untrusted network*, which can in principle have infinite computing power.
– Alice has a hand-held device H, which belongs to the untrusted network.
– Alice has a personal *decryption display* D consisting of an (LCD) display equipped with an additional array of light sensitive sensors [3] and some hardware to do a minimal amount of computations. Furthermore, it holds the same secret key K as TC.

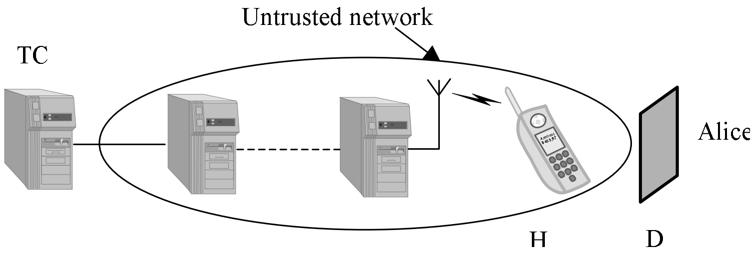

Fig. 1. *Communication between Alice and* TC *over an untrusted network.*

Alice wants to communicate with TC through her hand-held device H (e.g. PDA, mobile phone). The security of the solution that we propose makes use of the fact that Alice has two displays at her disposal (the display in H and decryption display D). TC will send only visually encrypted messages (using shared key K), to Alice. Alice's decryption display D will then generate the complementary randomised message according to the same key K. The message sent by TC will only be readable by Alice when she puts D on top of the display of H, on which the random pattern generated by TC appears. Note that the decrypted message is never available in electronic form.

2.2 The Problem

It will be the main topic of this paper to build a *Passive Two-Way Secure Authenticated Communication* channel ($PTWSAC$) in an untrusted environment using the model components of Section 2.1.

Definition 1. *A* PTWSAC channel *is a channel that allows for the following operations:*

[2] In some applications, TC can take the form of a trusted proxy [5].
[3] Light sensitive sensors can be easily and cheaply embedded in the pixels of an LCD display without reducing its picture quality.

- *Two way message authentication: the messages sent by* TC *are authenticated as well as the messages sent by Alice.*
- TC *can send secret messages of its choice to Alice (privacy).*
- *Alice can reply secretly to messages of* TC *but cannot send arbitrary messages of her own choice. This last property refers to the adjective 'passive'.*

Our results extend the visual cryptography system developed by Naor and Shamir in [11] and the protocols by Naor and Pinkas in [12], where transparencies are used to visually decrypt messages that are sent over an insecure channel. It was already recognised by the authors of [11] that such a system is only secure when each transparency is used only once, which makes their system rather impractical for general purposes.

3 A New Visual Crypto System with Maximal Contrast and Resolution

3.1 Naor-Shamir Visual Cryptography

Visual cryptography was first introduced in [11] and further investigated in [12, 15, 17]. The basic idea is to split a (black and white) image into two shares (visual encryption) which are printed on transparencies. The original image is reconstructed when the two shares are superimposed (Fig. 2). This process is called visual decryption. The Naor-Shamir approach suffers from a number of

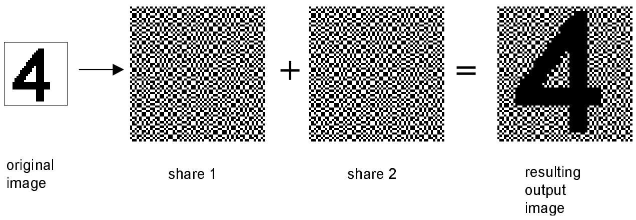

Fig. 2. *Naor-Shamir visual cryptography example where every original pixel is split into four sub-pixels.*

drawbacks: i) reduced resolution and contrast, ii) a secure system for multiple messages requires the use of a large number of transparencies, iii) Alice has to verify that the presented share on the untrusted terminal is a true random pattern [4], iv) not very well suited for coloured pictures

[4] This is to prevent an adversary from faking a message m by displaying a non-randomised version of m as a first share.

3.2 Visual Cryptography Using Polarisation Properties of LCD Displays

In order to solve the problems with the Naor-Shamir approach, we propose to use the polarisation properties of light. A first proposal in this direction was made in [2]. Our idea on the other hand is to use an LCD display as a polarisation rotator. This allows easy key management and efficient reconstruction of gray-scale and colour messages.

An LCD display consists of three main parts: a bottom polariser, an LC layer (consisting of small individual LC cells) and a top polariser. Polarisers project the polarisation of the incoming light into one direction (e.g. horizontally). An LC cell rotates the polarisation of incoming light depending on the voltage applied to that cell.

The visual crypto system we propose, consists of the following components (Fig. 3): a first LC layer (LC1) with a polariser on the bottom but **not** on top and a second LC layer (LC2) with a polariser on top but **not** on the bottom. The LC layers form the shares of this visual crypto system.

Since a picture consists of pixels, we will explain the visual crypto scheme for a single pixel. In order to obtain a scheme for the total image, the single pixel scheme must be repeated for all the pixels in the image. We start with a construction for black-and-white images (Fig. 3) of which we show a demo in Fig. 4.

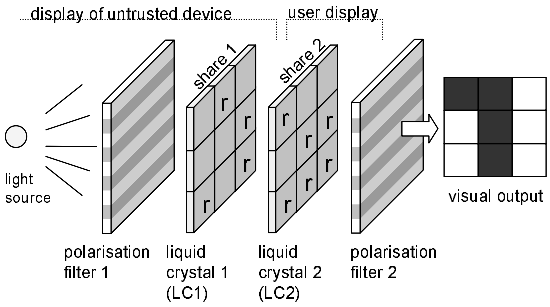

Fig. 3. *Visual cryptography by superposition of two liquid crystals. Cells in the liquid crystal layers that are indicated with an 'r' rotate the polarisation of light by $\pi/2$ radians.*

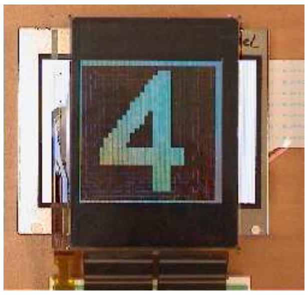

Fig. 4. *A hardware implementation that demonstrates the reconstruction method of Fig. 3.*

Black and White Pixels. Incoming light from the light source (e.g. backlight) contains light waves with all possible polarisations that lie in the plane perpendicular to the propagation direction of the light beam. The first polariser (Fig. 3) blocks the vertical components and after the polariser all light is horizontally polarised. A liquid crystal cell to which no voltage is applied (indicated by 'r' in Fig. 3) rotates the polarisation of the incoming light by $\pi/2$. An LC cell to which a certain voltage is applied does not change the orientation of the polarisation. Depending on the polarisation of the light leaving the second LC cell, the last polariser generates a black or white output pixel. LC2 can hence determine the output intensity independently of the polarisation direction of light that is leaving from LC1. Consequently, overlaying two LC shares behaves like an XOR-operation, in contrast to Naor-Shamir visual cryptography where overlaying the shares behaves like an OR-operation. General secret sharing schemes that can be built with this XOR construction are presented in [16].

The shares of a pixel are generated as follows. Denote with α_1 and α_2 the angles of rotation of corresponding cells in LC1 and LC2 respectively. The total polarisation rotation of light that passes both LC cells equals $\alpha = \alpha_1 + \alpha_2$. Using the symbol I_{r} for the normalised intensity of the reconstructed pixel, we have

$$I_{\mathrm{r}}(\alpha) = \cos^2 \alpha = \cos^2(\alpha_1 + \alpha_2) \ . \tag{1}$$

The decryption display D sets the angle α_2 to a value from the set $\{0, \pi/2\}$ according to the key K. As TC has the same key K (and hence knows α_2), he computes α_1 from the set $\{0, \pi/2\}$ such that $\alpha = \pi/2$ if the original pixel is black and $\alpha = 0$ or $\alpha = \pi$ if the original pixel is white.

It follows from the previous analysis that our scheme has the following advantages: i) Optimal resolution and contrast (since no sub-pixel patterns are used), ii) Elegant key updating mechanism by using the available electronics, iii) Alice does not have to verify the randomness of the image at H, since a non-randomised share produces a randomised reconstructed image. iv) Gray-scale and colour images can be efficiently reconstructed (shown below).

Gray Scales and Colours. The use of active LC layers allows efficient reconstruction of images with gray scales and colours. The LC cells in Fig. 3 can rotate the polarisation over an arbitrary angle within a certain range, say $[0, \pi/2]$ or $[0, \pi]$, depending on the construction of the LC and the applied voltage over an LC cell. In this case (1) holds with $I_{\mathrm{r}} \in [0, 1]$. Thus, by varying the value of α (or values α_1 and α_2) it is possible to change the intensity (gray scale) of a reconstructed pixel.

In order to implement visual cryptography with gray scales, D sets α_2 to a value from the interval $[0, \pi]$ according to the key K. TC has the same key K (and hence knows α_2) and computes α_1 such that the intensity I_{r} (Eq. 1) of the reconstructed pixel is equal to the intensity I_{O} of the original pixel. Since α_2 is chosen in $[0, \pi]$, α_1 has to belong to the interval $[0, \pi]$ due to the π-periodicity of I_{r} in order to reveal no information. If we assume that $\alpha_1, \alpha_2 \in [0, \pi]$, then α_1 can be determined by Algorithm 1.

Algorithm 1

INPUT: $I_{\mathrm{O}} \in [0, 1]$, $\alpha_2 \in_R [0, \pi]$.
OUTPUT: $\alpha_1 \in [0, \pi]$ such that $\cos^2(\alpha_1 + \alpha_2) = I_{\mathrm{O}}$.
1) compute $x = \arccos(|\sqrt{I_{\mathrm{O}}}|)$; $\eta \in_R \{x, \pi - x\}$.
2) if $\eta - \alpha_2 < 0$ then return $\alpha_1 = \eta - \alpha_2 + \pi$; exit.
3) if $\eta - \alpha_2 \geq 0$ then return $\alpha_1 = \eta - \alpha_2$; exit.

The idea of gray scales described above can be extended to colours. One colour pixel (Fig. 5) is built from three sub-pixels each of which has a different colour 'back-light' (Red, Green and Blue) by applying a colour filter. As with gray scales, the intensity of each of the colours can be changed individually by changing the rotations α_R, α_G and α_B of the red, green and blue colour respectively. In this way, any colour can be generated.

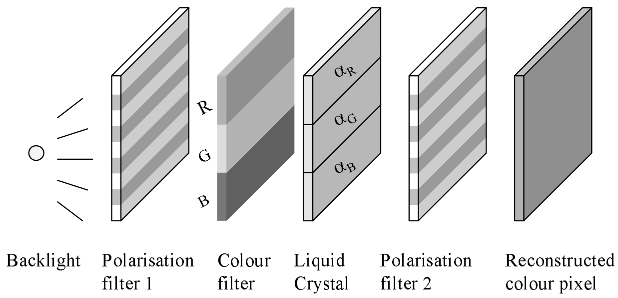

Fig. 5. *Schematic construction of a transmissive colour pixel in a standard LCD display.*

By applying Algorithm 1 three times per pixel, once for R, G and B, respectively, we can implement a colour visual cryptography system, without losing resolution. This stands in contrast to Naor-Shamir visual crypto systems as proposed in [13] and [17].

In a practical implementation, a pixel intensity can not have any value in $[0, 1]$ but is limited to a discrete set of k distinguishable values (e.g. $k = 256$). With k possible values for each colour component (R, G and B) a total number of k^3 colours can be displayed.

4 Building a PTWSAC Channel

4.1 Channels

In order to describe attacks on the system, we make a distinction between the communication from $TC \rightarrow$ Alice and the communication from Alice $\rightarrow$ TC.

The main communication from $TC \rightarrow$ Alice will be based on the visual crypto system explained in Section 3.2. The light sensitive sensors in D provide TC with an additional low-bandwidth return channel to Alice, used for key synchronisation (see Section 5.1). The display of H receives from TC a set of voltages that has to be applied to the different LC cells the display consists of. Attacking this communication channel therefore means either trying to guess the message that is hidden in the set of voltages (secrecy) or trying to change the voltages (authentication).

The communication from Alice $\rightarrow TC$ makes use of a reverse channel established by presenting to Alice in the reconstructed image a keypad/ keyboard. Alice can then reply to messages from TC by clicking on the appropriate positions in the reconstructed keypad and the clicked positions are communicated back to TC. Attacking this communication line means either changing those positions in a clever way to fool Alice and TC or trying to guess the message that Alice sends to TC.

4.2 Authentication from $TC \rightarrow$ Alice

In Section 3.2 is explained how TC can encrypt images that can be decrypted by Alice. This message is secret by construction but not authenticated.

An impersonation attack is very difficult since the visual crypto system is a secret key system with a key length equal to the number of pixels which in practice is of the order of at least 100 bits. Therefore, the success probability of an impersonation attack by guessing the correct key can considered to be negligible[5].

In visual cryptography a substitution attack mainly consists of adding, deleting or modifying characters, which makes this attack a bit more subtle. Eliminating this attack in a strong sense is impossible but not necessary: an attacker can easily change one (or only a very few) pixels while being undetected and without changing Alice's interpretation of the message. Therefore, loosely spoken, we state the following requirements for a visual cryptography authentication protocol:

- An attacker is not able to add or remove characters that make sense to humans.
- An attacker is not able to change characters into characters of his choice.

For a precise and more formal definition of visual authentication, we refer to Appendix A.

The substitution attack problem is solved by the following procedure for messages from $TC \rightarrow$ Alice.

- The background of the images sent from $TC \rightarrow$ Alice consists of a pattern with many colour transitions. The colours are randomly chosen and the pattern is placed at a random position.
- Each character sent by TC is depicted in one single uniform randomly chosen colour.
- The message is shown on a random place on the screen.

If an attacker wants to change a character into another one, he has to know where the character is positioned in the image and add or remove an area of uniform colour. In order to derive an upperbound for the probability of a successful substitution attack, we assume that the background contains a pattern

[5] Note that in a practical implementation, the security actually depends on the effective key length of the used Pseudo Random Number Generator (PRNG).

of randomly coloured grid cells, see for example Fig. 6.a. In this example the background is filled with triangular shaped grid cells (each grid cell consists of multiple pixels) on which the letter 'T' is displayed. Further we assume that the attacker knows the background *pattern* (i.e. where the colour transitions are) but not the specific *colours* of the grid cells. We denote by $\hat{c}$ the number of *partially* covered grid cells in the background of a shape (e.g. a character) that is to be removed by an attacker. With c we denote the number of *partially or completely* covered background grid cells by a shape to be added by an attacker (see Fig. 6.b). Note that in the example of Fig. 6, $\hat{c} = 33$ and $c = 38$.

The upperbound PU_{rem} for the probability of successfully removing an area of uniform predefined colour and the upperbound PU_{add} for the probability of successfully adding an area of uniform predefined colour are then given by

$$PU_{\text{rem}} = \left(\frac{1}{k^3}\right)^{\hat{c}} \quad \text{and} \quad PU_{\text{add}} = \left(\frac{1}{k^3}\right)^{c} . \tag{2}$$

We remind the reader that k^3 stands for the number of possible colours (see Section 3.2). The proofs of formulae (2) can be found in appendix A.

In general the system will be set up such that the attacker does not even know the background pattern nor the exact location of it (because it will be chosen differently for every message). Hence, Eqs. (2) are upperbounds for the probability of a successful substitution attack.

The protocol requires that Alice checks the colour uniformity of the characters and whether the background has the required format. This implies that in order to check authenticity of a message, Alice is in control herself.

a b

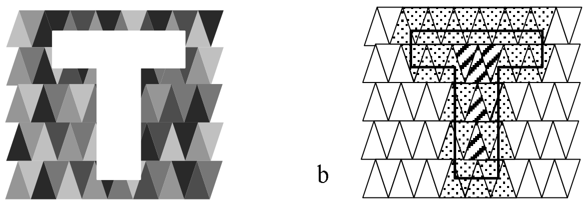

Fig. 6. *(a) Example of the letter 'T' on a multi-coloured background using 5 gray scales. (b) The 33 partially covered grid cells (dotted) and 5 completely covered grid cells (striped) in this example.*

We notice that when light sensitive sensors are present in each of the pixels of the decryption display, then the integrity of the communication channel from TC to the display of H can be based on message authentication codes (MAC) [10]. However, additional error correction codes need to be applied in order to be robust against pixel failures.

4.3 Secret and Authentic Channel (SAC) from Alice $\rightarrow$ *TC*

In this section, we will show how Alice can secretly reply in an authenticated way to messages from TC. We start with secrecy. In order to establish a private channel between Alice and TC, TC will send a visually encrypted message containing a picture of a randomised keypad/ keyboard (Fig. 7). The keypad is randomised in the sense that the characters are permuted in a random way but also the shape of the keypad might be randomised. This is done each time the keypad is sent to Alice. Moreover, we allow that a single character is displayed multiple times.

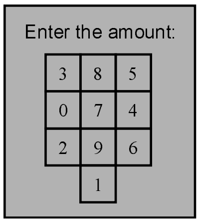

Fig. 7. *Example of a randomised keypad.*

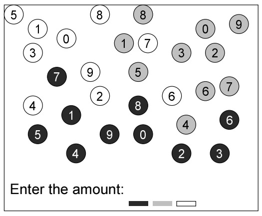

Fig. 8. *Example of various coloured keypads using three gray levels.*

Alice enters a number (e.g. an amount) by clicking appropriate *locations* in the image. The coordinates of the locations selected by Alice are sent back to TC. An eavesdropper will get no information on the message because the locations of the characters are random and hence unknown to the eavesdropper which does not have the secret visual decryption key.

Although the above mentioned system establishes a private channel from Alice $\rightarrow$ TC, the replies are not yet authenticated. An attacker can for instance easily perform a *swap* attack. This means that he flips the positions of some of the characters (positions) that are sent back. It is clear that this attack has a high probability of being undetected by TC. More generally, he can insert new positions which have a non-negligible probability of being accepted by TC.

A straightforward way to eliminate the swap attack is by adding a *verification* protocol. In this protocol TC uses the authenticated channel from TC to Alice (explained in Section 4.2) to transmit the message that it has received from Alice together with a confirmation code. In order to confirm that TC indeed received what she had sent, Alice has to type in the confirmation code. TC checks whether Alice confirmed the message with the correct code. The verification protocol leads to an authenticated channel from Alice $\rightarrow$ TC at the price of extra interaction.

Another way of eliminating the attacks mentioned above, is given by the following protocol. Instead of sending one randomised keypad, TC sends various keypads, each in a different colour (Fig. 8) to Alice. Alice enters the different symbols of her message in the colours indicated by TC. In this way the *swap* *attack* is prevented. Denote with w the area of an indicated colour and with A the total display area (where a pixel is taken as the unit of area). Then the probability of performing a successful substitution attack becomes proportional to $\frac{w}{A}$ per symbol (with a proportionality factor smaller than 1). In order to reduce this probability further, Alice can be asked to type her message multiple times ($l > 1$) with different colour indications. In that case the success probability becomes proportional to $\left(\frac{w}{A}\right)^l$. The use of colours can be avoided by using different shapes instead.

5 Application

5.1 A Practical Problem: Key Synchronisation

Since the visual crypto system, developed in Section 3, acts like a one-time pad, both TC and Alice need synchronised keys. In order to avoid large databases containing the keys, we use pseudo random number generators, $\mathrm{PRNG}(x, K)$ where x represents the state of the PRNG and K is the personal key that is secretly stored, in both TC and D. Hence, TC has to store for each user also the message key K_i which represents the current state of the user's PRNG [6].

In order to set up flexible key-synchronisation, we use the additional array of light sensitive sensors embedded in the pixels. Let K_i be the current key at TC and let h be a one-way hash function. TC uses K_i to visually encrypt the current message. At the bottom of the visually encrypted message, $h(K_i)$ is encoded in an optical signal that is modulated in the pixels of the complementary array [7]. D reads the encoded optical signal by using its light sensitive sensors. D performs the following algorithm, with K_j denoting the current key in D.

Algorithm 2

INPUT: $h(K_i)$.
MEMORY: (K_j, j) with $j \leq i$.
OUTPUT: update memory with (K_i, i), or no update.
0) $x := K_j$ and $J := j$.
1) while $h(x) \neq h(K_i)$ and $J - j < N$ do [8]
2) $x := PRNG(x, K)$ and $J := J + 1$
3) od.
4) if $J - j < N$ then $memory := (x, J)$ else no update.

The algorithm updates D's key K_j until it is synchronised with the key of TC. The maximum number of possible attempts is given by N. We require that D's

[6] Alice's security display D can be used in multiple applications by storing multiple personal keys (one for each application). Another option is implementing a proxy [5] (in that case TC represents Alice's proxy) such that only one personal key needs to be stored in the decryption display. The proxy will set up a secure communication to the application of Alice's choice.

[7] In order to avoid false detection of hashed keys, we can include a CRC check.

[8] If we replace this check by the requirement that $h(x)$ and $h(K_i)$ have a small distance to one another, then the algorithm is more robust against pixel failures.

PRNG is only executed in algorithm 2. Hence, $j \leq i$ is an invariant of algorithm 2. Furthermore, *TC*'s PRNG can only increase i such that $j \leq i$ is also an invariant of the protocol. In the case of a replay attack or a random message by an attacker, algorithm 2 will not make an update. After having performed algorithm 2, D can decrypt the message.

5.2 Communication Protocol

Based on the techniques developed in the previous sections, we describe the complete communication protocol.

Initialisation In order to initialise the system, Alice receives her decryption display D from *TC*. The PRNG in the display, D, is initialised in a certain state. This state is also secretly stored by *TC* together with the identity of the person it belongs to. A communication session starts with the following steps:

1. Using H, Alice notifies *TC* that she wants to start a communication session (by passing also her identity to *TC*).
2. *TC* looks up the PRNG and its state corresponding to the identity in its database.

Basic protocol We describe the basic protocol for establishing a PTWSAC.

1. Using their common key K_i, *TC* visually encrypts a message in an authenticated way as explained in Section 4.2.
2. Alice puts her trusted device D on top of H. D performs synchronisation algorithm 2. Alice checks whether the message sent by *TC* is authentic.
3. Alice answers the message of *TC* by clicking at the right positions in the keypad. She follows the protocol rules as developed in Section 4.3.
4. H sends the positions of the clicks to *TC* over the untrusted network.
5. *TC* interprets the received locations and possibly sends a verification message.
6. Alice replies on the verification message, if necessary.

6 Conclusions

In a world where information is an important commodity and should be available any time and anywhere, the integrity and confidentiality of the information and the security and privacy of the communication are important issues. In such a world the information will be presented in a user-friendly way, for example in graphical form on a display. However as devices like PDA's and cell phones are becoming more complicated and interconnected, they can no longer be trusted.

We present a solution for verifying the authenticity and confidentiality of information presented on untrusted displays based on visual cryptography. We propose some important practical technological modifications to the basic idea of visual cryptography leading to secure two-way communications. We replace transparencies with a small and cheap transparent LCD display. This enhances picture quality and allows for reconstructing colour images. The details of the visual crypto system presented in Section 3.2, are formulated as a mathematical model and are published in [16].

In order to extend the current solution to larger size displays, one can also make use of *flexible displays*. These are usually not made of Liquid Crystals. It will be investigated whether those materials offer interesting properties for the implementation of visual cryptography.

Acknowledgement: The authors like to thank the Oxygen project at MIT and M.T. Johnson for the stimulating discussions and G.J. Destura and Philips MDS Heerlen for the help with the demo.

References

1. M. Benham, *Internet Explorer SSL Vulnerability*, article on Thoughtcrime.org, August 2002, (see `http://www.thoughtcrime.org/ie-ssl-chain.txt`).
2. E. Biham, *Visual Cryptography with Polarisation*, Security Seminar , University of Camebridge, August 1997, (see `http://www.cl.cam.ac.uk/Research/Security/seminars/1997/`).
3. C. Blundo, A. De Santis and D. Stinson, *On the contrast in visual cryptography schemes*, Manuscript 1996. Available at `ftp://theory.lcs.mit.edu/pub/tcryptol.96-13.ps`.
4. M. Burnside, D. Clarke, B. Gassend, T. Kotwal, S. Devadas, M. van Dijk, R. Rivest, *The untrusted computer problem and camera-based authentication*, Springer LNCS2414, 2002.
5. M. Burnside, D. Clarke, T. Mills, A. Maywah, S. Devadas, R. Rivest, *Proxy based security protocols in networked mobile devices*, Proceedings SAC, 2002.
6. Case Western Reserve University, PLC group, *virtual textbook on liquid crystal displays*, `http://abalone.cwru.edu/tutorial/enhanced/files/textbook.htm`
7. J. Claessens, *Analysis and design of an advanced infrastructure for secure and anonymous electronic payment system on the Internet*, Ph.D. thesis, K.U. Leuven, December 2002.
8. T. Matsumoto, *Human identification through an insecure channel, Theory and Application of Cryptographic techniques*, 1991, 409-421.
9. T. Matsumoto, *Human-Computer Cryptography: An attempt*, ACM conference on Computer and Communication security, 1966, 68-75.
10. A. Menezes, P. van Oorschot, S. van Stone, *Handbook of Applied Cryptography*, CRC Press Inc., 1997.
11. M. Naor, A. Shamir, *Visual Cryptography*, Eurocrypt 94', Springer-Verlag LNCS Vol. 950, 1-12.
12. M. Naor, B. Pinkas, *Visual Authentication and Identification*, Crypto 97.
13. V. Rijmen, B. Preneel, *Efficient colour visual encryption or 'shared colors of Benetton'*, Presented at the rump session of Eurocrypt '96. Also available at `http://www.esat.kuleuven.ac.be/ rijmen/vc/` .
14. G. Simmons, *A survey of information authentication*, in Contemporary Cryptography - The science of information integrity, IEEE Press, 379-419.
15. D.R. Stinson, *An introduction to visual cryptography*, presented at Public Key solutions '97. Available at `http://bibd.unl.edu/ stinson/VCS-PKS.ps`.

16. P. Tuyls, H.D.L. Hollmann, J.H. v. Lint, L. Tolhuizen, *Polarisation based Visual Crypto System and its Secret Sharing Schemes.* Available at the IACR Cryptology ePrint Archive, `http://eprint.iacr.org/2002/194/`.
17. E. Verheul, H. van Tilborg: *Constructions and properties of k out of n visual secret sharing schemes*, Designs Codes and Cryptography, 11, 179-196, 1997.

A Visual Authentication

A.1 Definitions

By a visual authentication protocol the following is understood.

Definition 2 (Naor). TC *wishes to communicate to Alice an information piece m, the content of which is known to the adversary.*

1. TC *sends a message c to Alice which is a function of m and some shared secret information.*
2. *The adversary might change the message c before Alice receives it.*
3. *Upon receiving a message c', Alice outputs either FAIL or $\langle ACCEPT, m'\rangle$ as a function of c' and her secret information.*

In order to define the security requirements of a visual authentication system, we first note that the requirement that an adversary cannot convince Alice to receive any message different from the original message is much too strong. The change of one innocent pixel in the background for instance will with high probability not be detected by Alice. Therefore, we will only require that an adversary can not erase or enter symbols of his choice.

Definition 3. *Assume that Alice has the capabilities that are required from her for the protocol, that she acts according to the protocol and that the visual authentication system has the property that when the adversary is faithful, then Alice always outputs $\langle ACCEPT, m\rangle$. Let Σ be a set of messages that make sense to Alice (i.e. that she can understand). We call the system $(\Sigma, (1-p))$-authentic if for any message m communicated from* TC *to Alice, the probability that Alice outputs $\langle ACCEPT, m'\rangle$ where $m' \in \Sigma$ is at most p.*

The definition implies that it is hard for an adversary to change a message m into a message $m' \in \Sigma$ that makes sense to Alice. The set Σ depends on the application. In order to prove authenticity in specific applications the set Σ has to be defined carefully.

A.2 Substitution Attack

In order to derive an upper bound on the probability of a successful substitution attack we make the following assumptions. The background of an image sent by the trusted party contains a more or less regular grid, for example as depicted in Fig. 6. Every grid cell (a triangle in Fig. 6) of this grid gets a uniform colour chosen randomly from the k^3 possible colours. We further assume that the attacker knows the exact position of the grid but not the colours of the grid cells. The goal of the attacker is to introduce a shape on this background of uniform colour. The size of this shape covers, completely or partially, c of the grid cells.

Recalling the mechanisms of visual cryptography based on polarisation (see Section 3.2), it is clear that an attacker can change the colour of a grid cell by changing the rotations of all the pixels in a grid cell with the same amount. However, because he does not know the secret key of Alice, he does not know to *which* colour he is changing the grid cell. Thus, if he wants to change the grid cell to a predefined colour, his success probability is $\frac{1}{k^3}$.

If we finally assume that pseudo random processes generating keys, colouring of the grid cells by the trusted party, etc. are truly random, then we can state the following lemma concerning the addition of a uniform shape.

Lemma 1. *Suppose an adversary wants to introduce a shape of uniform predefined colour covering c grid cells and that he knows the location of the grid cells but not the colour. Then, the attackers success probability is upper bounded by $\left(\frac{1}{k^3}\right)^c$.*

Proof. In order to get a uniform, predefined colour in the added shape, every (part of a) grid cell that is covered by the shape should be changed to the predefined colour. The probability to obtain the right colour for one grid cell equals $\frac{1}{k^3}$. Therefore, the probability of changing all the c grid cells to the correct colour is $\left(\frac{1}{k^3}\right)^c$. □

Clearly, when the uniform colour of the added shape is not predefined, the probability becomes $\left(\frac{1}{k^3}\right)^{c-1}$.

By a similar reasoning we can derive an upperbound for the success probability of removing a shape. Here we make the same assumptions as above on the attacker's knowlegde on the position of the grid and the grid cells but we also assume that the attacker knows the exact location of the shape he wants to remove. In this case the probability depends on the number of grid cells which are *partially covered* by the shape. Changing a partially covered grid cell in one of uniform (arbitrary) colour, can be done with probability $\frac{1}{k^3}$.

Lemma 2. *Suppose an adversary wants to remove a shape of uniform colour partially covering $\hat{c}$ grid cells and he knows the location of the grid cells but not the colour. Then, the attackers success probability is upper bounded by $\left(\frac{1}{k^3}\right)^{\hat{c}}$.*

Proof. Grid cells that are completely covered by the shape to be removed can be easily turned into grid cells of a uniform colour by rotating their polarisations over an arbitrary angle. Therefore these pixels do not have to be taken into account. In order to be successful, an attacker has to change the colour of a partially covered grid cell, into a grid cell of a uniform but arbitrary chosen colour. Since the grid cell is partially covered his success probability is $\frac{1}{k^3}$ as explained above. Because he has to do this for $\hat{c}$ grid cells, his success probability becomes $\left(\frac{1}{k^3}\right)^{\hat{c}}$. □