How to Print a Secret

Aleks Essex
University of Ottawa

Jeremy Clark
University of Waterloo

Urs Hengartner
University of Waterloo

Carlisle Adams
University of Ottawa

ABSTRACT

We consider the problem of how to print a human-readable message, or image, on a piece of paper, while simultaneously preventing the participating printing agents or devices from learning its contents. We examine the problem in two settings: with a trusted dealer who knows the message, and in a distributed scenario that allows two non-colluding printers to obliviously generate the secret message and print it without ever learning it. We present a basic protocol for printing arbitrary-length messages in the trusted dealer model, as well as protocols for printing different types of messages in the distributed model: a randomly selected element from a set, a random permutation of elements in a set, and an optimization for printing alphanumeric characters using 16-segment display logic.

1. INTRODUCTION

The next time you find yourself printing a document such as this, consider how you would go about it if, for some reason, you did not want your printer to know the contents. In this paper we consider the problem of printing written material (text, graphics, etc) on paper in a human-readable format, without the printing agents or devices learning its contents.

The main results of this paper (besides introducing this new paradigm of printing) include a scheme for printing arbitrary-length messages using a trusted dealer and multiple non-colluding printers in section 3, distributed schemes for two non-colluding printers to randomly and obliviously select and print a single element from a set of elements in section 4 and a random permutation on a set of elements in section 5. We present an optimization of the protocols for alphanumeric printing using 16-segment display logic in section 6. More generally we note that these protocols could be applied both as a form of distributed visual cryptography, as well as to other types of media beyond the single-sheet manifestation primarily envisioned herein.

2. PRELIMINARIES FROM RELATED WORK

Visual Cryptography.

Visual cryptography (VC) is a specialized secret sharing scheme due to Naor and Shamir, where a secret image is split into a number of shares and each share is printed on a transparency [9]. When the shares are recombined, by layering them overtop of each other, the secret is restored. There can be any number of shares, and the threshold of shares required to recover the secret can be set at any appropriate value. Pertinent literature to this work is the perceptual effects of misaligned shares [8], the use of seven-segment displays with VC [2], and the application of VC to electronic voting [4].

Although the schemes presented here could be distributed across an arbitrary number of VC shares (and hence printers), for simplicity we consider a basic form of VC with two shares only. Consider a secret image s as an $m \times n$ matrix of pixels, where each pixel is either 0 for transparent or 1 for opaque. The first share α is generated by randomly selecting a 0 or 1 for each pixel. This share is then `XOR`ed with the secret image to generate the second share β. Thus, the original can be reconstructed by $s = \alpha \oplus \beta$. However, printing α and β on their own sheet of transparency paper and stacking them is equivalent to an `OR` operation, not an `XOR`.

To correct for this, the basic VC scheme maps each pixel in α and β into a 2×2 block of sub-pixels, which we call a VC-pixel. Without loss of generality thus map is defined as □→□■ ■□ and ■→■□ □■. By layering VC pixels, we get either a fully opaque VC-pixel, defined as a 1, or a half-transparent VC-pixel, defined as a 0. This emulates the exclusive-or operation where: ■■=■□+□■=■□+□■, while □■=□■+■□and ■□=□■+□■.

Invisible Ink.

The term "invisible ink" is used to describe a class of inks that, in their initial state, are transparent and un-pigmented (*i.e.,* "invisible"), but contain a color-forming chemical that, when placed in contact with a developing agent (or process), become opaque and darkly pigmented (*i.e.,* visible).

Simple invisible inks can be derived from common acidic household substances, such as lemon juice or vinegar. A solution of such a substance, heavily diluted in water, can be manually applied to paper in the form of text using classical ink-well era writing implements (*e.g.,* quill, fountain pen, etc), or more generally any pointed object (*e.g.,* paper clip, stylus, etc). This clear solution, once dried, can be developed (*i.e.,* activated) by subjecting the paper to a moderate heat source (*e.g.,* iron, oven at low temperature, candle, etc), thereby revealing the hidden text. Alternatively, these types of inks can be viewed under black light.

Recent work due to Chaum *et al.* has undertaken to develop invisible inks for verifiable optical-scan election ballots [3]. In this application the inks were specifically designed to be both amenable for use in commercial-off-the-shelf ink-jet printers, as well as activated by a developer chemical placed

into a special-purpose pen (as opposed to a heat source). In its most basic form, one of the color channels of a color inkjet printer is replaced with an ink cartridge containing the invisible ink. Graphical objects rendered entirely in this color channel, when printed, are produced in invisible ink instead.

Preliminary technical improvements to the indistinguishability of invisible ink, especially under black light, employ randomized overprinting of variably ultraviolet-reactive (but non-activating) inks to effectively camouflage the message when viewed under an assortment of wavelengths of light.[1] This process most recently has been expanded to consider the CcMmYK (*i.e.,* 6 color channel) model facilitating a wider assortment of camouflaging agents, protecting against a wide range of passive eavesdropping tactics.

Document Authentication.

If a document has been given to someone, and after some time it is returned, document authentication can provide assurance that the returned document is the same as the original document and not a forgery. Document authentication involves extracting a unique set of features of the document that can be tested for at a later time. Ideally, the features are robust to the document being somewhat mishandled. For physical documents, features can include paper color, paper texture, and ink splatter. Recent work due to Clarkson *et al.* proposes a practical scheme based on fuzzy feature extraction from the three dimensional shape of the paper [6]. This fingerprinting scheme can be implemented using a commodity scanner and is robust against additional ink being printed on the paper, as well as light mishandling of the document. As part of the following schemes we will require either the dealer or the recipient to be able to perform some form of document authentication.

3. PRINT AN ARBITRARY-LENGTH SECRET USING A DEALER

General Model.

A dealer D wants to have an arbitrary-length secret printed on a sheet of paper, intended for a recipient R, by a third party. D instructs two non-colluding entities offering print service, Printer A and Printer B, on how to print an image of a secret without either printer learning the secret.

Motivating Example.

Consider the case where a bank, D, wants to distribute credit card numbers and activation codes to a large set of customers, R, through the mail. Due to the volume, the bank must outsource the printing to a printing service, Printer A, but is concerned that these secret numbers may be surreptitiously compromised. Instead, T would like to distribute the trust between two printers so that both printers would have to be compromised, or collude with each other, to learn the secrets.

Solution.

The core of our proposed solution is a visual cryptography scheme in which individual shares are printed by separate, non-colluding, printers in succession onto a single sheet of paper using invisible ink, as shown in Figure 1.[2] This approach differs from the original VC proposal, which suggests printing shares on cellulose acetate (*i.e.,* overhead) transparencies, which can be aligned on top of a non-transparent share, such as a computer monitor or piece of paper, to reveal the secret [9]. The advantage of employing a single-sheet approach over that of a multi-sheet approach is threefold.

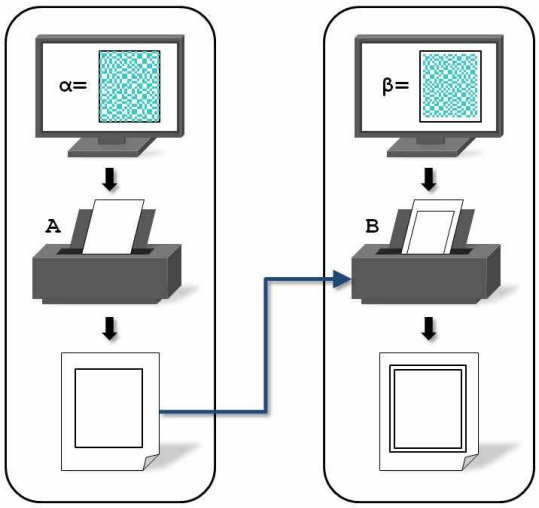


Figure 1: Distributed printing. Printers A and B receive separate visual crypto shares of a message. Printer A prints its share in invisible ink and passes the sheet onto Printer B who, in turn, prints its share on top.

- **Alignment:** With VC, proper alignment (or registration) is necessary to reconstruct the message. Misalignments less than a sub-pixel reduce the contrast of the message, while greater misalignments render it unreadable [8]. In our single-sheet scheme, alignment of the shares is a matter of attention for the printers, not the recipient. Presumably an industrial printing process is better suited to guarantee proper share alignment than the recipient.
- **Usability:** Our single-sheet approach offers a simpler user experience. The role of visual cryptography is not central to the recipient recovering the message and arguably, the recipient could be completely unaware of it.[3]
- **Fewer Chains of Custody:** Perhaps most important, physical separation of shares need not be enforced prior to the recipient receiving them. Assuming the

[1]Private communications with Scantegrity project. In particular, R. Carback and D. Chaum. 2008-9.

[2]While the last printer could print its share in non-invisible ink, we assume that each share is printed in invisible ink. This prevents the last printer's share from being learned if the paper is observed by earlier printers after being fully printed.

[3]Our scheme may require the recipient to have a revealing pen. Other applications for our work include optical-scan voting, where codes can be printed in the ovals marked by the voter [3]. Here the use of a revealing pen directly substitutes the use of a normal pen when marking a ballot.

scheme is secure within the privacy model described below, then the recipient will receive decisive feedback (i.e., tamper evidence) if the secret was viewed prior to its receipt.

As the primary application of invisible inks are in the transport of secret messages, in general we consider two threats to message secrecy:

- **Passive exposure**: A message written in invisible ink becomes temporarily visible in a particular environment (*e.g.,* ultraviolet light), to an optical sensor (*e.g.,* eye, camera, *etc.*).
- **Active exposure**: A message written in invisible ink is rendered permanently visible by initiating a one-way chemical process that develops (*i.e.,* activates) pigmentation in the ink.

We seek to mitigate these threats by requiring an invisible ink printing process with the following characteristics:

- **Indistinguishability of undeveloped ink**: A message printed in invisible ink is said to be resistant to passive exposure if any two messages are indistinguishable.
- **Tamper evidence**: A message printed in invisible ink is said to be resistant to active exposure if any actively attacked message is easily distinguished from an untouched message.

A simple attack for eschewing indistinguishability and tamper-evidence is one where a malicious party actively exposes (*i.e.,* develops) the message, records it, and reprints it on a new sheet of paper. The key to preventing this line of attack is in establishing the authenticity of the paper sheet. Using a document authentication scheme such as that mentioned in Section 2, either the dealer, recipient, and/or the printers would seek to establish document authenticity at some time after the printing process.

In the working example above, the bank could keep an inventory of sheets issued to Printer A and could verify their authenticity after having the sheets returned by Printer B (it could also at this time reveal the invisible ink). Alternatively, the recipient could perform the verification upon receiving the paper. In the following sections concerning schemes that do not employ a dealer, Printer A will publish an inventory of the sheets issued to Printer B. Both printers can check the sheets against the inventory at any point in the protocol, and after Printer B has applied its shares, the sheets can be authenticated by the recipient. A final alternative is to use an honest-but-curious third party to provide this service. In all cases, the verification could be conducted through a random audit of a small portion of the sheets.

Definition 1. ***Invisible Ink Secrecy Model.*** *We say a message printed in invisible ink is physically secure in the invisible ink secrecy model if that message is indistinguishable under passive exposure, and tamper-evident under active exposure.*

Assuming the existence of an invisible ink printing system, secure in the invisible ink secrecy model, we propose a simple distributed printing scheme, outlined in algorithm 1, involving a trusted dealer D issuing two VC-shares of a secret to two printers: Printer A and Printer B. Printer A prints its share on a piece of paper in invisible ink. This sheet is given to Printer B, which prints its share in invisible ink, directly on top of Printer A's share. The resulting sheet is provided to a recipient R, who uses a special developing pen to reveal the combined shares (and hence the secret). See Figure 2.

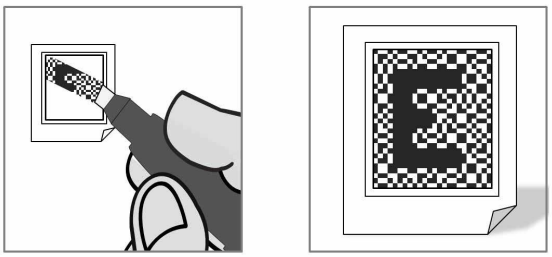


Figure 2: Visual Crypto in Invisible Ink. Left: A special revealing pen activates (darkens) the invisible ink. Right: A message printed as visual crypto shares, is revealed to the recipient.

Algorithm 1: Printing a secret with a trusted dealer

Dealer D's Private Input: Secret message $s \in [0,1]^{m \times n}$ as $m \times n$ monochrome pixel matrix

Dealer D should:
  Fingerprint sheet of paper p
  Choose $\alpha \in_R [0,1]^{m \times n}$
  Compute $\beta = s \oplus \alpha$
  Send α and β to Printer A and B respectively
end
Printer A should:
  Print α in invisible ink onto p
  Send p to Printer B
end
Printer B should:
  Print β in invisible ink on top of α on p
  Send p to Dealer D
end
Dealer D should:
  Authenticate paper p
  Send p to recipient R
end

4. PRINT A RANDOMLY SELECTED SECRET WITHOUT A DEALER

General Model.

Two non-colluding entities offering print service, Printer A and Printer B, randomly and obliviously select a message from a public set of possible messages. The pixel representation (*i.e.,* image) of the message is printed on a sheet of paper intended for a recipient R without the use of a dealer. The selected message is unknown to both Printer A and Printer B and will only be known by R. Furthermore, print-

ers A and B can each enforce the randomness of the selection independent of each other.

Motivating Example.

Consider the case where Printer A and Printer B want to print a random alpha-numeric character on a sheet of paper to give to R. By doing this successive times, they could print a multi-character string where each character is independently selected at random. A number of voting systems with cryptographic end-to-end integrity require ballots to be printed with various codes. One example is Scantegrity [3], which requires secret and random codes to be printed beside each candidate and already uses invisible ink to hide the value of the codes. Another is ThreeBallot [10], which requires a set of statistically unique identifiers to be printed on each ballot. Printers or poll-workers who observe these identifiers, prior to the ballot being cast, threaten ballot secrecy.

Protocol.

Let the set of possible messages be S and of order $N = |S|$. Consider, as in the motivating example above, that $S = \{A \dots Z, 0, \dots, 9\}$ in which each $s_i \in S$ shall be represented as an m by n monochrome pixel matrix that visually expresses it. At a high-level, Printer A will generate a random visual crypto image , α, as her share, and print it onto the paper in invisible ink. She will then, for each message $s_i \in S$, generate the complementary set of shares for Printer B: $\beta_i = \alpha \oplus s_i$, for $1 \leq i \leq N$. She randomly permutes the order of the set, obfuscates each element, and sends the set to Printer B. Printer B then selects $\gamma \in_R [1, N]$ to be deobfuscated, where $\in_R$ denotes the uniform random selection of an element from a set. He then prints β_γ over top of α in invisible ink.

We require three additional properties of this distributed protocol in the absence of a dealer, namely:

i. Printer A should not learn which visual crypto share β_γ was selected by B.
ii. Printer B should not learn the value of any share β_i other than the single share, β_γ, he selected.
iii. Printer B should not know which message s_i corresponds to β_γ.

We utilize a 1-out-of-N oblivious transfer protocol to achieve properties (i) and (ii). s_i is perfectly hidden by Printer A's share α. Thus (iii) holds under the assumption of non-collusion of the printers.

The full details of the protocol are provided in Algorithm 2. The oblivious transfer sub-protocol is due to Tzeng [11], which we selected for its reusable public parameters and minimal message exchange (2-pass). It is set in the ring of integers modulus a large prime p, with multiplicative subgroup of prime order q. By using a Pedersen commitment in line 4, Printer B's choice of share is perfectly hidden ensuring (i). Property (ii) holds because line 24 for an arbitrary j reduces to $\beta_j h^{r_j(\gamma - j)}$ allowing the recovery of β_j only when $\gamma = j$.[4] Property (iii) holds because $s_i = s_{\pi^{-1}(j)} = \beta_j \oplus \alpha$, and Printer B does not know α or random permutation $\pi()$.

[4]Security remark: If printer B could easily recover β_j for $j \neq \gamma$, then Printer B could easily perform an arbitrary discrete logarithm: either $\log_g(g^{r_j})$ to recover β_j directly or $\log_g(h)$ to find x' such that $g^{x'} h^j = y$. Concerning the latter, it is thus important that g, h are generated by Printer A or through a distributed key generation (DKG) protocol.

Algorithm 2: Printing a single secret character with oblivious transfer

Public Parameters: Set of alphanumeric characters S, and primitive roots $g, h \in \mathbb{G}_q$

1 **Printer B should:**
2 Choose index to select: $\gamma \in_R [1, N]$
3 Choose random secret: $x \in_R \mathbb{Z}_q^*$
4 Commit to private choice: $y = g^x h^\gamma$
5 Send y to Printer A
6 **end**
7 **Printer A should:**
8 Perform lines 2, 3, and 8 from Algorithm 1
9 **for** $1 \leq i \leq N$ **do**
10 Select i^{th} message from set: $s_i \in S$
11 Compute complimentary share: $\beta_i = \alpha \oplus s_i$
12 Randomly permute index: $\beta_i \rightarrow \beta_{j=\pi(i)}$
13 Choose random secret: $r_j \in_R \mathbb{Z}_q^*$
14 Compute: $c_j = \langle a_j, b_j \rangle = \left\langle g^{r_j}, \beta_j (\frac{y}{h^j})^{r_j} \right\rangle$
15 Send p and set of c_j's, ordered by j, to Printer B
16 **end**
17 **Printer B should:**
18 Flip coin $c = \{H, T\}$ with $\Pr[H] = \rho$
19 **if** $c = H$ **then**
20 Request α, β_j, and r_j, $\forall j$, from Printer A.
21 If correct, repeat protocol from Line 7.
22 **else**
23 Select c_γ
24 Compute: $\beta_\gamma = \frac{b_\gamma}{(a_\gamma)^x}$
25 Print: β_γ on top of α on paper p
26 **end**
27 **Recipient should:**
28 Authenticate paper p
29 **end**

Printer A could misconstruct the set of β_j values such that when they are combined with α, they do not each produce a unique character (*e.g.,* any selection by Printer B will result in the same character being printed). Given property (ii), Printer B could not detect such an attack directly. Thus Printer B shall, with some probability ρ, perform a cut-and-choose audit of Printer A's construction of α and β_j values to ensure they are properly formed.

An additional feature of the protocol is that both parties contribute to the *random* selection of $s_i \in S$. Printer A chooses a random permutation $\pi : i \rightarrow j$, and Printer B selects a random index γ to print. Thus if one of the two parties select messages deterministically, the contribution of the other will be to ensure the printed message is, in fact, randomly selected.

This protocol outlines how to print a single, secret, random, and obliviously selected alphanumeric character on a piece of paper. It is easy to see that the oblivious transfer could be conducted several times, independently, to produce a string of characters for applications such as those offered above as motivating examples.

5. PRINT A PERMUTATION OF A SET OF MESSAGES WITHOUT A DEALER

General Model.

Two non-colluding entities offering print service, Printer A and Printer B, randomly select a permutation and apply it to a public set of possible messages. The pixel representations (*i.e.,* images) are printed on a sheet of paper intended for a recipient R without the use of a dealer. The order of these images is unknown to both Printer A and Printer B and will only be known by R. Furthermore, printers A and B can each enforce the randomness of the permutation independent of each other.

Motivating Example.

A number of voting systems with cryptographic end-to-end integrity require ballots to be printed with a randomized candidate ordering. Prêt-à-voter [5] and Aperio [7] require candidate names to be listed on the ballot in an independent random order. The candidate list could be developed immediately prior to voting to increase voter privacy. Alternatively, consider a contest where the names of prizes are printed in invisible ink on a set of tickets. For example, a batch of one dozen tickets could be printed as follows: ten shall say "please play again" while the other two shall each name a different prize. By applying a random permutation to these twelve strings, not even those running the contest will know for certain which tickets contain a prize.[5]

Solution.

A permutation of a set of N images requires N elements to be printed. In its most basic form the 1-out-of-N oblivious transfer (see algorithm 2) is run N times, with Printer A selecting N independent, random, visual crypto shares α. However instead of applying independently selected permutations $\pi_1, \cdots, \pi_N$ at each successive execution, the same random permutation π_1 is applied to S when constructing the complementary set of VC shares $\{\beta_{(1,k)}, \cdots, \beta_{(N,k)}\}$ during the k-th execution.

However since a permutation of elements requires every element to be appear once and only once, we shall require a mechanism to enforce non-repetition of elements. Such non-repetition of VC shares constructed by Printer A can be made through a similar cut-and-choose process as that mentioned in section 4. However to enforce non-repetition of the selections made by Printer B, we extend algorithm 2 by algorithm 3 such that Printer B proves the uniqueness of her selections, γ_k, without revealing the order of selection, as well as providing Printer A with the ability to perform a cut-and-choose on Printer B's selections.

As a brief description of algorithm 3, Printer B constructs N commitments $y_k = g^{x_k} h^{\gamma_k}$ and sends them, along with the sum of random factors $\hat{x}$ to Printer A. We index each message in the set using a public set of indices, selected from a superincreasing sequence κ (*e.g.,* $\{k \in \mathbb{Z} : \kappa_k = 2^k\}$). Thus, if Printer B selects the same index more than once, it is impossible to adjust the other selections such that they sum to $\hat{\kappa}$ and are valid indices. Printer B could select an index not in κ, however this would forfeit him from learning at least one proper share. This will be caught by the cut-and-choose subprotocol in lines 12-15, which can be thought of as an optional step for scenarios where a malicious printer could get away with misprinting a share.[6] To verify that each of the commitments y_k contains a unique index choice, Printer A will calculate their product $\hat{y}$ causing the exponents γ_κ to sum. Given public parameter $\hat{\kappa}$ and Printer B's assertion $\hat{x}$, Printer A will verify whether $\hat{y} = g^{\hat{x}} h^{\hat{\kappa}}$ thus verifying B's honesty in selecting each element once.

Algorithm 3: Printing a secret permutation with oblivious transfer

Public Parameters: Superincreasing indices $\kappa = \{\kappa_1, \cdots, \kappa_N\}$ that sum to $\hat{\kappa}$.

1 **Printer B should:**
2 **for** $1 \leq k \leq N$ **do**
3 Choose, without replacement, index: $\gamma_k \in_R \kappa$
4 Choose random secret: $x_k \in_R \mathbb{Z}_q^*$
5 Commit to private choice: $y_k = g^{x_k} h^{\gamma_k}$
6 Send y_k to Printer A
7 Compute $\hat{x} = \sum_{k=1}^{N} x_k$ and send to Printer A
8 **end**
9 **Printer A should:**
10 Compute: $\hat{y} = \prod_{k=1}^{N} y_k$
11 Verify: $\hat{y} = g^{\hat{x}} h^{\hat{\kappa}}$
12 Flip coin $c = \{H, T\}$ with $\Pr[H] = \rho$
13 **if** $c = H$ **then**
14 Request γ_k, x_k, and y_k, $\forall k$, from Printer B.
15 If correct, repeat protocol from Line 1.
16 **end**
17 **for** $1 \leq k \leq N$ **do**
18 Run Algorithm 2 at line 8.

6. EFFICIENTLY PRINT TEXT

General Model.

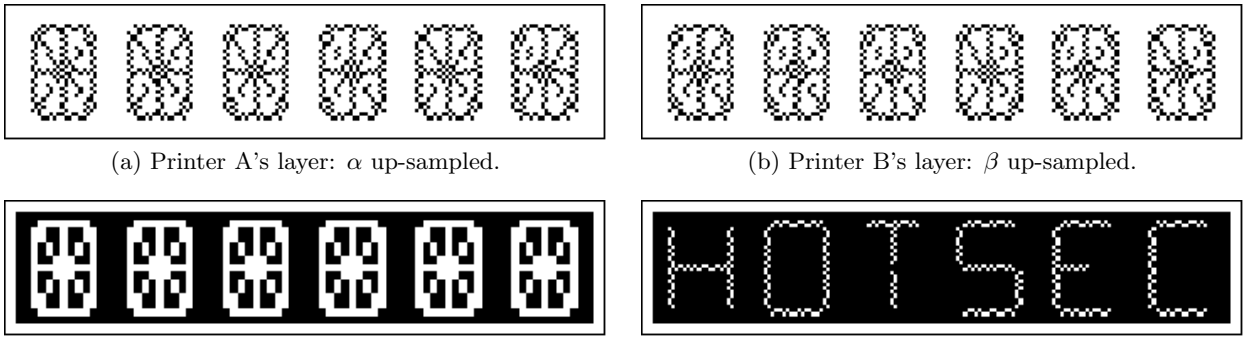
Two non-colluding printers, Printer A and Printer B, randomly select a short string of alphanumeric characters from a public set of possible strings. The model has the same properties as the general model in Section 4 but is optimized for the use of alphanumeric strings. It can be used in conjunction with Algorithms 2 and 3.

Motivating Example.

An official is to issue a survey that contains a question about sensitive information that respondents may not answer honestly for fear of retribution. A mitigating technique is randomized response, where a sensitive question can be replaced with its negation in a random fraction of the surveys [1]. Thus the issuers do not learn any particular respondent's true response with certainty, but can statistically adjust the results to estimate the number of respondents who answered in a particular way. Consider the problem of

[5]This could prevent documented cases of fraud, such as http://archives.cnn.com/2001/LAW/08/21/monopoly.arrests/

[6]For example, in the voting scenario, the permutation will be observed by the voter to be correct before it is used, thus not requiring this optional step. However it may be desirable for printing tickets in a contest.



(a) Printer A's layer: α up-sampled.

(b) Printer B's layer: β up-sampled.

(c) Static background mask.

(d) The three layers combined.

Figure 3: Visual Crypto Up-sampling. Example of a string of text using 16-segment up-sampling, comprised of two visual crypto layers set against a static background. Instead of representing each segment as a collection of traditional 2×2 visual crypto pixels, each segment's share can be transmitted as a single bit and then 'up-sampled' to an arbitrarily large, public, and pre-arranged visual crypto pixel pattern for improved perceptual clarity. In this way visual crypto shares can be fully expressed by 16 bits per alphanumeric character, regardless of perceptual resolution.

obliviously printing a survey, where a particular question is randomly selected from a set that contains nine elements of question Q and one instance of question ¬Q. The answers are returned on a different sheet of paper (*e.g.,* a Scantron form), and the question sheet is discarded by the respondent after using it. The mechanism in this section can also be used to optimize the motivating examples in Section 5: for printing candidate names in voting systems or the names of prizes in a contest.

Solution.

There is an upper-limit to the size of the message that can be transferred in one instance of the protocol: in this case, it is the security parameter of the system, which is likely to be 1024 or 2048 bits. For the purpose of efficiency, our motivation is to encode as many characters as possible into one ciphertext payload.

We defined messages to be a monochrome pixel matrix of dimensions $m \times n$. Consider, for example, the image of a character to be 26 by 18 sub-pixels that are either white or black (the resolution that will be used in Figure 3). These parameters would require 117 bits per character, allowing just over half a dozen characters to fit into a single ciphertext payload. Increasing the resolution (and hence the readability) comes at the cost of using additional encryptions to convey the same content. To improve on this, we can use segment displays. In a segment display, alpha-numeric characters are displayed by driving a subset of 16 segments, or 7 segments if we restrict ourselves to numbers. At 16 bits per character, we can fit over 60 alpha-numeric characters into one 1024-bit secret: enough to encode a short question, candidate name—regardless of the character resolution.

With the use of Algorithm 2, Printer A can obliviously transfer one of many message shares to Printer B, where the share (β_γ) is a sequence of 16-bit segment encodings. If the exclusive-or of this sequence is taken with Printer A's accompanying sequence (α), the result is the character encoding of a randomly selected string (s_γ). However, we deviate from Algorithm 2 in that Printer A and B cannot print α and β_γ directly. Instead Printer A and B must up-sample their 16-segment sequences into a VC share.

The process of up-sampling works as follows. A pixel matrix of any dimension is partitioned into segments and background; white and black in Figure 3(c) respectively. To aid with perception, the segment portion (white portion) is filled with random-looking VC pixels (recall a VC pixel is a 2x2 sub-pixel representation of a pixel). This same pixel mask can be used for every character. Printer A up-samples α by taking the pixel mask and for every segment, either leaving the segment as is if α is 0 for that segment or flipping all the bits in the segment if α is 1. See Figure 3(a). Printer A prints this and the background mask in invisible ink. Similarly Printer B up-samples β using the same pixel masks, generates a VC share in Figure 3(b), and prints it.

7. CONCLUSION

We have demonstrated an interesting, novel paradigm: selected messages can be printed on a sheet of paper without the printers learning them. We have outlined a number of scenarios where such a property may be useful, including password distribution, cryptographic voting, contests, and randomized response surveys. Indeed these protocols may be relevant to other applications of visual cryptography, in particular those requiring a dealer-less solution, as well as to other types of document and display media. We hope this work points to new possibilities now that we know how to print a secret.

8. REFERENCES

[1] A. Ambainis, M. Jakobsson and H. Lipmaa. Cryptographic Randomized Response Techniques. *Public Key Cryptography (PKC)*, 2004.

[2] B. Borchert. Segment-based Visual Cryptography. *WSI Tech Report (WSI-2007-04)*, 2007.

[3] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. Rivest, P. Ryan, E. Shen, and A. Sherman. Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes. *Electronic Voting Technology Workshop (EVT)*, 2008.

[4] D. Chaum. Secret-Ballot Receipts: True Voter-Verifiable Elections. *IEEE Security & Privacy*,

2(1), 2004.

[5] D. Chaum, P. Y. A. Ryan, and S. Schneider. A practical voter-verifiable election scheme. *ESORICS*, 2005.

[6] W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J.A. Halderman, and E.W. Felten. Fingerprinting Blank Paper using Commodity Scanners. *IEEE Symposium on Security and Privacy*, 2009.

[7] A. Essex, J. Clark, and C. Adams. Aperio: High Integrity Elections for Developing Countries. *Workshop on Trustworthy Elections (WOTE)*, 2008.

[8] F. Liu, C.K. Wu, X.J. Lin. The alignment problem of visual cryptography schemes. *Designs, Codes and Cryptography*, 50(2), 2009.

[9] M. Naor and A. Shamir. Visual Cryptography. *EUROCRYPT*, 1994.

[10] R. Rivest and W.D. Smith. Three Voting Protocols: ThreeBallot, VAV, and Twin. *Electronic Voting Technology Workshop (EVT)*, 2007.

[11] W.G. Tzeng. Efficient 1-Out-of-n Oblivious Transfer Schemes with Universally Usable Parameters. *IEEE Transactions on Computers*, 53(2), 2004.