Oblivious Printing of Secret Messages in a Multi-party Setting

Aleksander Essex and Urs Hengartner

Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 2G1
{aessex,uhengart}@cs.uwaterloo.ca

Abstract. We propose oblivious printing, a novel approach to document printing in which a set of printers can cooperate to print a secret message—in human or machine readable form—without learning the message. We present multi-party protocols for obliviously printing a secret in three settings: obliviously printing the contents of a ciphertext, obliviously printing a randomized message, and generating and obliviously printing a DSA/Elgamal keypair. We propose an approach to improving the legibility of messages in the presence of numerous participants. Finally we propose some potential applications of oblivious printing in the context of electronic voting and digital cash.

1 Introduction

Since the days of Gutenberg the privacy model for document printing has been the same: a printer must learn the content of a message in order to print it. In this paper we take a fundamentally new approach to printing, one in which a human- or machine-readable message can be printed *without* the printers learning its content.

We believe oblivious printing can useful in a variety of real-world situations where it may advantageous to receive a secret in printed form. As an example, consider a scenario in which a user needs to receive a secret, but lacks access to the appropriate software, hardware or network infrastructure, such as in certain mobile or financial settings. Another potential scenario might be one in which a user needs to create a secret but does not understand how, or is otherwise unmotivated to take the proper steps to so securely, such as in the creation of strong passwords. Oblivious printing might even be useful when a user's computer cannot be trusted to handle a sensitive computation, such as in the case of internet voting. We describe several concrete applications later in the paper.

The Oblivious Printing Model. Oblivious printing is a protocol in which a group of printers cooperate to print a secret message. This message can be revealed and read by the intended recipient, but remains unknown to the printers. Oblivious printing is accomplished through a combination of cryptographic and document security techniques. The high level procedure is sketched as follows:

1. Message selection: Printers execute a secure multi-party protocol to select a message (under encryption) from an alphabet of valid messages.
2. Graphical Secret Sharing: Printers convert the message (under encryption) into a graphical image. Using a dealerless protocol they secret share the pixels between themselves.
3. Invisible ink overprinting: Pixel shares are converted into a visual crypto pattern. Using invisible ink each printer successively prints their share on the same sheet of paper and in a known location/orientation.
4. Message recovery: The recipient of the completed document activates the invisible ink of the combined shares (e.g., using a special activation pen), thereby revealing the message.

We presented a preliminary two-party protocol for oblivious printing of randomized messages based on oblivious transfers [10]. The techniques presented in this paper generalize the model to a fully multi-party setting. Additionally this approach allows for the secret message to be simultaneously output as an obliviously printed document and as an associated ciphertext allowing greater possibilities for integration into broader protocols.

Contributions and Organization. In this paper we present the oblivious printing paradigm and give three novel multi-party protocols: in Section 3 we present a protocol for obliviously printing the contents of an encrypted message, in Section 4 we present a protocol for obliviously printing a randomized message with improved contrast over the first protocol. We then present an extension to the second protocol for generating and obliviously printing an Elgamal/DSA keypair. In Section 5 we suggest a possible method for mitigating contrast drop-off as the number of printers increases based on the existence of AND-ing invisible inks. Finally in Section 6 we suggest some possible applications of oblivious printing for trustworthy electronic voting and electronic cash.

2 Preliminaries

2.1 Physical Security

Printing is ultimately a physical process, which means that any oblivious printing scheme will have a physical security component to it. In this paper we assume ideal security properties although we acknowledge in practice they can be challenging and costly to implement and difficult to guarantee.

Invisible ink. Invisible ink is an ink that, as its name implies, is initially invisible when printed. The ink becomes visible (i.e., pigmented) after it is activated. Ideal invisible ink has two security properties,

– Invisibility: Messages printed in invisible ink should be unreadable prior to activation,

– Activation-evident: Activated ink should always be plainly evident to anyone viewing the document.

Work has been done in developing invisible inks in the context of trustworthy optical-scan voting as part of the Scantegrity II system [5]. Ballots with confirmation codes printed in invisible ink were recently fielded in a live municipal election in the United States [4]. For the sake of our description we assume that there exists an ink printing process with the above properties.

Document Authentication. Techniques for determining a document's authenticity are an important component of oblivious printing. Ideally document authentication can efficiently and definitively distinguish between authentic and non-authentic (counterfeit) documents.

Anti-counterfeiting methods (e.g., watermarks, holographic foil, embedded magnetic strips, etc) exist but can be cost-prohibitive. It was shown by Buchanan et al. [3] that fiber patterns can be used to uniquely identify paper documents. Clarkson et al. [8] later developed a paper fiber identification method using commercial-grade scanners. Sharma et al. [24] implement a paper fingerprinting scheme based on texture speckles using a USB microscope costing under $100.

For the sake of our description we assume that there exists an efficient scheme for determining a physical document's authenticity.

2.2 Visual Cryptography

A visual cryptography scheme (VCS) is a visual secret sharing scheme in which a (secret) message or graphical image is split into a number of shares. An early example of visual secret sharing is due to Kafri and Keren [16] (what they call "random grids"), although Shamir and Naor [18] are generally credited with the paradigm in the security literature. The latter outline a collection of visual crypto schemes for which the shares of some threshold $k > 2$ out of n printers are necessary to recover the image and is denoted as (k, n)-VCS. Ateniese et al. [2] generalize this notion to access structures for which the message is recoverable under arbitrarily defined subsets of participants. A survey of a number of variations of visual cryptography is presented in [28].

Optimal Contrast of an (n, n)-VCS. An image is secret shared by a trusted dealer on a pixel-by-pixel basis. To share a pixel, the dealer issues each printer a unique and randomly assigned pattern of sub-pixels chosen to enforce the desired access structure. Shamir and Naor [18] prove the optimal number of sub-pixels for an (n, n)-VCS is 2^{n-1}. In this scenario if the dealer wishes to share a black pixel, the shares are constructed such that when an authorized set of printers combine their shares, each of the resulting 2^{n-1} sub-pixels will be black. Similarly if the dealer wishes to share a white pixel, one of the resulting sub-pixels will be white (the other $2^{n-1} - 1$ will be black). This is used to define a measure of contrast, α, as being the relative difference in intensity between the combined

shares resulting from a white pixel and a black pixel in the original image. The optimal contrast for an (n, n)-VCS is thus $\alpha = \frac{1}{2^{n-1}}$.

Visual Crypto as Used for Oblivious Printing. We make use of some aspects of visual cryptography for the purposes of oblivious printing; however there are several important differences with how it is typically presented in the literature:

1. Invisible ink shares: Printers successively overprint their shares in invisible ink on a **single sheet of paper**. Activation of the combined invisible ink shares recovers the message.
2. Dearlerless share creation: The message is distributed to shares by a multi-party computation.
3. Fixed sub-pixel patterns: Each printer has a fixed pair of sub-pixel patterns. Which of the two patterns the printer ends up printing is secret, but the patterns themselves are a public input to the protocol.

We will make use of a set of sub-pixel patterns that implement an XOR operation. Work has been done into visual cryptography in a variety of physically XOR-ing media including interferometers [17], light polarization [25], and even image reversal using modern photocopy machines [27]. In our approach, however, the XOR is approximated in an underlying physical medium (i.e., over-printed shares of invisible ink) that implements an OR.

Definition 1. *An n-input visual XOR, n-VCX, describes a set of sub-pixel patterns that visually implement an XOR of the input bits in a physically OR-ing medium.*

Let S be an $n \times 2^{n-1}$ binary matrix for which each column is unique and has an even Hamming weight. Let $\overline{S}$ be the element-wise complement of S. Let sub-pixel pattern matrix Φ be as follows: $\Phi(l, 0) = S(l, :)$ and $\Phi(l, 1) = \overline{S(l, :)}$.

For a set of Boolean values $a_1 \dots a_n \in \{0, 1\}$ and their associated logical exclusive-or $a' = \bigoplus_{i=1}^{n} a_i$, we say the sub-pixel pattern matrix Φ implements an n-input visual crypto exclusive-or, if the sub-pixel pattern produced by overlaying shares $\Phi(1, a_1) \dots \Phi(n, a_n)$ has the following outcome: the total number of black sub-pixels is $2^{n-1} - 1$ if $a' = 0$, and respectively 2^{n-1} when $a' = 1$. If the a_i's contain an even number of ones (i.e., the XOR is zero), then exactly one of the columns will end up with all 0's (i.e., a white sub-pixel) due to the way the matrix was designed and the pixel will be visually interpreted as white. If the a_i's contain an odd number of ones (i.e., their XOR is one), all columns will contain a non-zero number of 1's due to the way the matrix was designed and the pixel will be visually interpreted as black. Φ implements an n-VCX.

Example 1. A 4-VCX: Let inputs $[a_1, a_2, a_3, a_4] = [1, 0, 0, 1]$ and,

$$S = \begin{bmatrix} 0\,0\,0\,1\,0\,1\,1\,1 \\ 0\,0\,1\,0\,1\,0\,1\,1 \\ 0\,1\,0\,0\,1\,1\,0\,1 \\ 0\,1\,1\,1\,0\,0\,0\,1 \end{bmatrix}.$$

We have $\Phi(1,1) = [1,1,1,0,1,0,0,0]$, $\Phi(2,0) = [0,0,1,0,1,0,1,1]$, $\Phi(3,0) = [0,1,0,0,1,1,0,1]$, and $\Phi(4,1) = [1,0,0,0,1,1,1,0]$. When the vectors are ORed, it produces the sub-pixel pattern $[1,1,1,1,0,1,1,1]$. Such a pattern is visually interpreted as intended, i.e., a white pixel with contrast $\alpha = \frac{1}{8}$.

3 Obliviously Printing an Encrypted Message

In this section we present a protocol for obliviously printing the contents of a ciphertext for which the associated plaintext is within a known, bounded, alphabet of m possible valid messages. Given an encrypted plaintext $[\![p]\!]$, a set of n printers $\mathcal{P}_1 \ldots \mathcal{P}_n$, each with a share of the decryption key, will jointly print p as a $(u{\times}v)$-pixel image I_p depicting p in a human- or machine-readable form such that no printer learns p. We leave the origin of $[\![p]\!]$ generic although we envision it as being the output of some other (previous) multi-party computation.

3.1 Translation Table

A translation table is defined in which each element is a valid possible message that can be printed and for which each message consists of an association between a plaintext value and a bitmap that depicts it. The translation table is taken as input to the protocol and is used to facilitate the translation of a message—under encryption—from its plaintext form to its bitmap depiction.

Let translation table T consist of m message pairs representing the set of valid messages that can be printed. Each message pair $\langle t, \mathrm{I}_t \rangle \in T$ consists of a plaintext value t in the plaintext domain, and a $(u{\times}v)$-pixel monochrome bitmap I_t depicting t as a human- or machine-readable image. Each value $\mathrm{I}_t(i,j) \in \{0,1\}$ corresponds respectively to a white or black pixel.[1] We use the notation $[\![T]\!]$ to denote the element-wise encryption of T. Each message pair $\langle t, I_t \rangle \in T$, can be regarded as a vector of $uv+1$ elements, $[t, I_t(0,0), \ldots, I_t(u{-}1, v{-}1)]$ where each element is encrypted separately. The initial encryption of each element is taken with a known random factor (e.g., 0). Mixing $[\![T]\!]$ involves re-randomizing each element and shuffling by the message pair vectors.

In order to facilitate mixing and searching for elements in $[\![T]\!]$ under encryption, $|T|$ in practice will be small relative to the plaintext domain. Because I_t will be encrypted at the pixel-level we note that for practical purposes the image size should be kept small. Using a technique described by Essex et al. [10], however, text can be optimized using segment-based display logic. A single alphanumeric character (or digit) can be fully described in 16 (respectively 7) encryptions regardless of the resolution of the visual crypto sub-pixel pattern used.

3.2 Setup

Let $\langle \mathsf{DKG}, \mathsf{Enc}, \mathsf{DDec} \rangle$ be an encryption scheme with distributed decryption. Distributed key generation $\mathsf{DKG}(\mathsf{n})$ generates a public key, e, and a private key share,

[1] Printing uses a *subtractive* color model and thus the plaintext values assigned to color intensities are the reverse of that found in the computer graphics literature.

d_i associated with printer $\mathcal{P}_i$. Encryption $[\![m]\!] = \mathsf{Enc}_e(m, r)$ is semantically secure and homomorphic in one operation. Distributed decryption $\mathsf{DDec}_{d_i}(c)$ of a ciphertext c is possible with all n printers applying their shares d_i. Without loss of generality we use Elgamal [20].

3.3 The Protocol

The protocol for obliviously printing a $p \in T$ given $[\![p]\!]$ is described in Protocol 1 and consists of Sub-protocols 1.1 and 1.2. Briefly, Sub-protocol 1.1 encrypts and mixes T and searches it (under encryption) for the entry corresponding to p, outputting the associated encrypted bitmap. The process of searching the encrypted translation table for a value and outputting the associated encrypted image as described in Step 2 of Sub-protocol 1.1 is closely related to the Mix and Match system [13]. In Step 1 of Sub-protocol 1.2 the printers secret share a pixel by homomorphically XOR-ing it with random bits in a manner similar to the technique used by Cramer et al. [9].

Finalization Layer. Given n printers note that Protocol 1 uses an $(n{+}1)$-VCX. An additional "finalization" layer allows the printers to verify the correctness of printing without ever revealing the message. For each pixel, each printer will generate a random bit, and using the sub-pixel pattern matrix, print it in invisible ink. A cut-and-choose proof is performed in Step 2 to demonstrate the printers correctly printed their random bits. Then an $(n{+}1)$-th finalization layer is computed by homomorphically XOR-ing the message bit with each of the random bits. Since the finalization layer is essentially a one-time pad, it can be decrypted without revealing the message. Finally, the finalization layer is printed using black ink, the correctness of which can be verified visually by inspection.

PROTOCOL 1 (Obliviously Print p given $[\![p]\!]$)

Input: Translation table T, encrypted plaintext $[\![p]\!]$, sub-pixel matrix Φ implementing an $(n{+}1)$-VCX, soundness parameter ρ.

Output: A document with a $(u \times v)$-pixel image depicting p, printed in invisible ink and with contrast $\alpha = \frac{1}{2^n}$.

The protocol:

1. Translate encrypted plaintext into associated pixel-wise encrypted image: Run Sub-protocol 1.1.
2. Obliviously print encrypted image: Run Sub-protocol 1.2.

SUB-PROTOCOL 1.1 (Translate $[\![p]\!]$ into $[\![\mathrm{I}_p]\!]$)

Input: Translation table T, encrypted plaintext $[\![p]\!]$.

Output: A $(u \times v)$ pixel-wise encrypted image of p (i.e., $[\![\mathrm{I}_p]\!]$).

The protocol:

1. Encrypt and verifiably mix translation table T: Each printer participates in a verifiable mix network, which encrypts and shuffles the message pairs $\langle t_i, \mathrm{I}_{t_i} \rangle \in T$. The result is denoted $[\![T']\!]$.
2. Find $[\![p]\!]$ in $[\![T']\!]$: printers search $[\![T']\!]$ attempting to locate a $[\![t_i]\!]$ for which $t_i = p$:
   (a) For each message pair $\langle [\![t_i]\!], [\![\mathrm{I}_{t_i}]\!] \rangle \in [\![T']\!]$, the printers perform a test of plaintext-equality between $[\![p]\!]$ and $[\![t_i]\!]$.
   (b) If a match is found, output the corresponding pixel-wise encrypted bitmap $[\![\mathrm{I}_{t_i}]\!]$. If no match is found the protocol terminates and an error message is output.

Remark: Various protocols exist for verifiable mix networks. One efficient and statistically sound technique for multi-column mixing is due to Sako and Kilian [23]. The plaintext equality test (PET) is due to Juels and Jakobsson [13].

3.4 Obliviously Printing an Arbitrary Plaintext

In Protocol 1 we showed how to obliviously print a plaintext $p \in T$ given its encryption. As was previously mentioned, in order to make mixing and searching $[\![T]\!]$ feasible, $|T|$ will typically be quite small relative to the plaintext space.

We briefly sketch how any message from the plaintext space might be accommodated. To print an arbitrary p, first the printers would define an alphabet Σ (e.g., the Latin alphabet) for which p could be represented as a string Σ^l. The printers would execute a multi-party pre-protocol to convert $[\![p]\!]$ into a collection of ciphertexts $[\![p_1]\!] \ldots [\![p_l]\!]$ for which $p = p_1 || \ldots || p_l$ (a multi-party protocol for extracting bit-fields under encryption is left to future work). The printers would then run Protocol 1 for each p_i, printing the result on the same sheet of paper.

4 Obliviously Printing a Randomized Message

In this section we present a contrast optimization in the special case where the printers are printing a randomized message $r \in_r T$. Although Protocol 1 can also be used for this purpose the protocol presented in this section has a contrast of $\alpha = \frac{1}{2^{n-1}}$ (as opposed to $\alpha = \frac{1}{2^n}$). Protocol 1 allows the printers to engage in a cut-and-choose proof of correct printing without revealing p directly. This is done at the expense of contrast: the use of the finalization layer introduces an additional layer forcing the n printers use an $(n+1)$-VCX, which has half the contrast relative to an n-VCX.

SUB-PROTOCOL 1.2 (Obliviously Print $[\![\mathrm{I}_t]\!]$)

Input: A $(u \times v)$ pixel-wise encrypted image $[\![\mathrm{I}_t]\!]$, sub-pixel matrix Φ implementing an $(n+1)$-VCX, soundness parameter ρ.

Output: A document with I_t printed in invisible ink with contrast $\alpha = \frac{1}{2^n}$.

The protocol:

1. Obliviously Print ρ instances of I_t: For each $1 \leq i \leq \rho$:
   (a) Print a new instance (sheet): For each pixel $[\![\mathrm{I}_t(j,k)]\!]$:
      i. Post commitments to random bits: Each printer $\mathcal{P}_{l \leq n}$ draws a random bit $b_{i,j,k,l} \in_R \{0,1\}$ and broadcasts a non-malleable commitment to it.
      ii. Secret share pixel: The n printers jointly compute an encrypted finalization pixel $[\![f_{i,j,k}]\!] = [\![t(j,k) \oplus b_{i,j,k,1} \oplus \ldots \oplus b_{i,j,k,n}]\!]$ using a partially homomorphic XOR.
      iii. Print sub-pixel pattern in invisible ink: Each printer $\mathcal{P}_{l \leq n}$ records the unique physical characteristics of the paper sheet and overprints the sub-pixel pattern $\Phi(l, b_{i,j,k,l})$ in invisible ink on the i-th document instance at the position associated with pixel (j,k).
2. Perform cut-and-choose proof of correct printing: The printers select $\rho - 1$ documents at random to audit (see remark). For each chosen sheet:
   (a) Prove:
      i. Unveil commitments: Each printer unveils their uv commitments generated in Step 1a-i).
      ii. Prove XOR: Each printer broadcasts their random factor used in computing the partially homomorphic XOR in Step 1a-ii).
      iii. Activate invisible ink: The printers collectively activate the invisible ink revealing the result of Step 1a-iii).
   (b) Verify: Each printer performs the following steps. If any of them do not hold, the protocol is terminated and an error message output:
      i. Check commitments: Verify commitments produced in Step 2a-i).
      ii. Check XOR: Recompute the homomorphic XOR using $[\![t(j,k)]\!]$ and the random factors revealed in Step 2a-ii) and confirm the result equals the finalization pixel generated in Step 1a-ii).
      iii. Check printing: For each pixel ensure the combined VC sub-pixel pattern created by the bits revealed in 2a-i corresponds to the printed version.
      iv. Check paper: Authenticate the sheet against those in Step 1a-iii.
3. Finalize the remaining sheet:
   (a) Decrypt finalization layer: The printers decrypt the finalization pixels $[\![f_{i,j,k}]\!]$.
   (b) Print finalization layer: The printers authenticate authenticate the sheet. If the sheet is not recognized, the protocol terminates and an error message is output. Without loss of generality $\mathcal{P}_1$ prints the finalization layer: each pixel $\Phi(n+1, f_{i,j,k})$ is printed in black ink at the associated position. The other printers check the finalization layer is printed correctly. The resulting document is securely delivered to its intended recipient.

Remark: A partially homomorphic XOR using exponential Elgamal is due to Neff [19]. The heuristic due to Fiat and Shamir [12] can be used to fairly select documents to audit.

If the message is randomized, then revealing it as part of a cut-and-choose process does not reveal information about the remaining (unactivated) messages. So instead of partially printing ρ copies of a single message p, auditing $\rho - 1$ copies and finalizing the remaining copy, the printers instead obliviously print ρ complete and independently random messages, of which they audit $\rho - 1$. The protocol is described in Protocol 2.

Arbitrary-length random messages can be built by repeated (independent) executions of Protocol 2 on the same sheet of paper, which may be useful in the creation of strong passwords, cryptographic keys or random tokens. Note in this setting the bit-field extraction step outlined in Section 3.4 would be unnecessary.

PROTOCOL 2 (Obliviously Print a Random $r \in_r T$)

Input: Translation table T, sub-pixel matrix Φ implementing an n-VCX, soundness parameter ρ

Output: A document with a $(u \times v)$-pixel image depicting a random $r \in_r T$, printed in invisible ink and with contrast $\alpha = \frac{1}{2^{n-1}}$. Encrypted plaintext $[\![r]\!]$.

The protocol:

1. OBLIVIOUSLY PRINT ρ INDEPENDENT RAND. MSGS.: For each $1 \leq i \leq \rho$:
   (a) SELECT RANDOM MESSAGE PAIR FROM T: Run Step 1) from Sub-protocol 1.1 to generate $[\![T'_i]\!]$. Without loss of generality the printers select encrypted message pair $[\![T'_i(0)]\!] = \langle [\![r_i]\!], [\![\mathrm{I}_{r_i}]\!] \rangle$.
   (b) OBLIVIOUSLY PRINT $[\![\mathrm{I}_{r_i}]\!]$: Run Step 1a) from Sub-protocol 1.2 with the following modifications:
   - Without loss of generality, the first $(n-1)$ printers $\mathcal{P}_{l<n-1}$ partially decrypt the secret shared pixel $[\![f_{i,j,k}]\!]$ created in Step 1a-ii) by applying their respective shares of the private key.
   - Similar to Step 1a-iii) each printer $\mathcal{P}_{l<n-1}$ overprints their VC sub-pixel pattern $\Phi(l, b_{i,j,k,l})$. Printer $\mathcal{P}_n$ decrypts the partial decryption of $[\![f_{i,j,k}]\!]$ and prints $\Phi(n, (b_{i,j,k,n} \oplus f_{i,j,k}))$ in invisible ink.

   (c) PERFORM CUT-AND-CHOOSE PROOF OF CORRECT PRINTING: The printers select and audit $\rho-1$ documents as in Step 2) of Sub-protocol 1.1.
   (d) OUTPUT REMAINING SHEET: The remaining document $\mathrm{I}_{r'}$ is securely delivered to its intended recipient. The associated ciphertext $[\![r']\!]$ is output.

4.1 Generating and Obliviously Printing a DSA Keypair

One interesting variation of Protocol 2 might be generating and obliviously printing an DSA/Elgamal keypair for which the printers do not know the private key. This could potentially be an interesting approach to building a PKI in which a group of printers acting as a distributed CA issues keypairs in physical form.

Our initial work [10] allowed for the oblivious printing of random strings, but could not construct the associated ciphertext. In this paper we can obviously print random strings for which we have the associated ciphertext from which we can compute the associated public key.

The keypair can be rendered in a convenient encoding such as alphanumeric (e.g., Base64) or 2-D barcode (e.g., a QR-code). We note that 2-D barcodes often contain additional error correction information. Creating a valid error-correction codes under encryption is something we leave to future work. We present a protocol for generating and obliviously printing a DSA/Elgamal keypair in Protocol 3.

PROTOCOL 3 (Generate and Obliviously Print an Elgamal Keypair)

Input: A large prime $p = 2\alpha q+1$ (for a small integer α), a generator $g \in \mathbb{G}_q$, an encoding alphabet Σ (e.g., Base64) for which $|\Sigma|$ is a power of 2.

Output: A document with public key $y = g^{sk}$ printed in black ink, and secret key sk printed in invisible ink.

The protocol:

1. For $0 \leq i < \left\lfloor \frac{\log_2(q)}{|\Sigma|} \right\rfloor$:
   (a) INITIALIZE T_i: For $0 \leq j < |\Sigma|$: Add message pair $\langle g^{j+|\Sigma|i}, I_{\Sigma(j)} \rangle$ to T_i.
   (b) OBLIVIOUSLY PRINT PRIVATE KEY SEGMENT: Printing on the same sheet each time so as to build a string of characters, run Protocol 2 with T_i as input, receiving an (encrypted) segment of the private key $c_i = [\![g^{sk_i}]\!]$.
2. RECOVER PUBLIC KEY: Printers decrypt $[\![y]\!] = [\![g^{\sum_i r_i}]\!] = \prod_i c_i$. Without loss of generality $\mathcal{P}_1$ prints the result in black ink and other printers confirm the value is correctly printed. The result is securely delivered to the intended recipient.

Remark: If the secret key's bit-length does not evenly divide the encoding alphabet, the above loop is run one final time with a reduced alphabet $\Sigma' \subset \Sigma$ where $|\Sigma'| = \log_2(q) \bmod |\Sigma|$.

5 Mitigating Contrast Drop-off with AND-ing Inks

Using the basic invisible ink described above we note that contrast declines exponentially in the number of printers. In practice this greatly limits the number of printers that can participate and still produce a legible message. Other factors like image size, resolution and font play a role in legibility but in general we would not expect an obliviously printed document to be legible with more than about half a dozen printers.

We have discussed invisible ink in the context of a physical disjunction (i.e., an OR). In that setting a pixel will darken on activation if any of the shares contain invisible ink. However it seems invisible ink printing could offer other possibilities if the pigmentation reaction could be customized to realize a different logical construction. We briefly examine the properties that can be achieved if it were possible to formulate invisible inks that implement a physical conjunction (i.e., an AND). Chemically it seems possible such inks could be formulated; the basic ink process as described throughout this paper (cf. [4]) already forms a type of chemically-based conjunction between the invisible ink itself and the activating substance. Granted it would likely be a challenge to formulate conjunctive inks that were invisible for more than a small k. We are not aware of the existence of such inks. It is worth noting, however, that if such inks *could* be formulated, they have the potential, at least in theory, to achieve optimal contrast (i.e., $\alpha = 1$) in the presence of arbitrarily many printers.

Definition 2. *A set of k inks are k-way conjunctive if, upon activation, a pixel darkens iff all k inks are physically present.*

We denote an n-VCX implemented with k such "AND-ing" inks as a (k, n)-VCXA. To create sub-pixel share matrix Φ in this setting we begin by constructing the $(n \times 2^{n-1})$ matrix S (refer to Definition 1) and then evenly segmenting it into $\frac{2^{n-1}}{k}$ sub-matrices of size $(n \times k)$. Each sub-matrix represents a sub-pixel, and each element in the sub-matrix instructs the printer whether to print the associated ink in that sub-pixel or not. Using this approach a (k, n)-VCXA has a contrast $\alpha = \frac{k}{2^{n-1}}$ (k is a power of 2 and the optimal contrast ratio remains $\alpha = 1$).

Example 2. A $(4, 4)$-VCXA: Let inputs $[a_1, a_2, a_3, a_4]$ and S be the same as in Example 1. The 4-way conjunctive inks are labeled A, B, C, and D. Each share instructs the printer which of the four inks to print in each of the two-subpixels. The shares are: $\Phi(1, 1) = [\{A, B, C\}, \{A\}]$, $\Phi(2, 0) = [\{C\}, \{A, C, D\}]$, $\Phi(3, 0) = [\{B\}\{A, B, D\}]$, and $\Phi(4, 1) = [\{A\}, \{A, B, C\}]$. The conjunction of the shares produces $[\{A, B, C\}, \{A, B, C, D\}]$. Since the first sub-pixel will not contain the ink D when the shares are printed, it will never activate. The second sub-pixel will contain all four inks when printed and therefore will darken when activated. The pixel therefore will contain one white and one black sub-pixel which is visually interpreted as intended, i.e., a white pixel with contrast $\alpha = \frac{1}{2}$. By comparison with Example 1 the contrast is 4x greater.

6 Example Applications

Electronic Voting. Cryptographically verifiable electronic voting is a natural application for oblivious printing. In this setting voters receive a receipt of their ballot that allows them to confirm their vote was correctly counted, yet without revealing it to anyone. A vital requirement of any secret ballot election employing the receipt paradigm is that no single party, *including* the ballot printer(s), may gain an advantage in deducing how a voter voted.

Printing Verifiable Optical-scan Ballots Voting by paper optical-scan ballot is a common method used in the United States [26] today. However work into cryptographically verifiable optical-scan voting (cf. [5, 6, 21]) has continued to entrust ballot printers with secret and identifying information. Recently in [11] we presented a two-party approach to obliviously printing ballots based on the preliminary techniques in [10]. Through this work, we can extend it to a fully multi-party setting—a feature long realized in fully-electronic proposals.

Multi-factor ballots for Internet Voting Internet voting has been a recent and popular topic of interest. One successful open-source and cryptographically-verifiable internet voting platform is Helios.[2] Helios accepts encrypted votes (along with zero-knowledge proofs of validity), which are then homomorphically tallied [1]. One fundamental and well-known limitation of this approach is that the voter's computer must be trusted to construct the encrypted ballot and is vulnerable to virus/malware. Using Protocol 3, encrypted Helios votes could be prepared on a voter's behalf and mailed to them on an obliviously printed ballot form. The voter would cast their vote by submitting the ciphertext corresponding to their candidate. Similarly, a verifiable internet voting scheme due to Ryan and Teague [22] proposes a multi-factor solution based on acknowledgment codes cards, which are mailed to the voter. The acknowledgment code cards contain secret information and so oblivious printing may of use here also.

Coercion-resistant internet voting Beginning with Juels et al. [15], work into coercion-resistant internet voting has attempted to extend privacy protection to voters, even when casting their ballots in an unsupervised environment. Clark and Hengartner [7] propose a coercion-resistant scheme based in part on an in-person registration protocol requiring voters to select secret passphrases and be able to (privately) compute randomized encryptions of them. Such passphrases and their encryptions could instead be pre-computed and obliviously printed by a distributed election authority, potentially simplifying the in-person registration phase and simultaneously enforcing higher-entropy passphrases.

Electronic Cash. Bitcoin[3] is an interesting recent proposal for digital currency. Transactions are timestamped and inserted into a common transaction history (known as a "block chain") using a proof-of-work model. An account consists of a DSA keypair: a private signing key is used for sending funds and a public key is used for receiving them. A transaction consists of two components. The first component points to an earlier transaction in the block chain in which funds were sent to the account corresponding with the user's public key (and for which the funds have not already been spent). The second component involves the user signing the transaction (which includes the destination account) using the private signing key. Typically these keys are stored on a user's machine in a "wallet" file. One interesting alternative is Bitbills,[4] a service which issues Bitcoins in physical

[2] `http://heliosvoting.org`
[3] `http://bitcoin.org`
[4] `http://bitbills.com`

form. A Bitbill consists of a plastic card (similar to a credit card) corresponding to a set amount of Bitcoins. The associated private signing key is printed on the card as a 2-D barcode and hidden under a tamper-evident/holographic covering. The funds can be redeemed in by scanning the card with a smartphone.

Importantly, knowledge of the private signing key is necessary and sufficient to transfer funds and recent criminal activity has focused on stealing such keys from users' computersas well as online Bitcoin bank accounts[5]. Therefore any currency issuing service like Bitbills would have to be trusted never to redeem the cards it issues, and to prevent any private keys to fall into the hands of hackers. Oblivious printing could be used to create a *distributed* currency issuing service. With Protocol 3 adapted to an elliptic curve setting, keypairs could be generated and printed without any individual issuer knowing the private key thereby enforcing that only the cardholder can redeem the funds.

7 Security Analysis

We briefly sketch some of the security properties of our system. For space reasons we limit our discussion to Protocol 1 (i.e., Subprotocols 1.1 and 1.2).

Cryptographic Security. Informally there are two security properties we seek for the cryptographic component of the protocol. One is *integrity*: a printer should be convinced that the combined shares depict an image of the (encrypted) input. The other property is *secrecy*: an adversary in collusion with a subset of printers should not be able to determine the input.

We assume the commitment function is non-malleable, hiding and binding. The assumptions regarding encryption are stated in Section 3.2. The completeness, soundness and secrecy of Sub-protocol 1.1 follow directly from [14] [13]. If the printers follow Sub-protocol 1.2 they will always produce a finalization layer that, when XOR-ed with the individual shares, recovers the input. Secrecy of the commitments and encryptions follow from the assumptions. Secrecy of the decrypted finalization layer follows if one or more printers select random bits. Soundness is probabilistic and follows from the cut-and-choose proof. The independence of the random bits is enforced by the non-malleable commitment function. Correct computing of the homomorphic XOR is established by the cut-and-choose proof when printers reveal their commitments and the random factors used to compute the XOR.

Physical Security. For simplicity we proceed with our discussion of physical security in a setting in which the printers receive their shares from a trusted dealer through a private and authenticated channel. In the physical setting we seek two security properties. One is *integrity*: a printer should be convinced that the combined printed shares match the combined received shares. The other property is *tamper evidence* which is closely related to secrecy: an adversary should not be able to determine the output of the protocol without corrupting all printers or tampering with the document, which will then be evident.

[5] `http://mybitcoin.com/archives/20110804.txt`

We assume the invisible ink can only be read in its activated state and that activated ink is plainly evident. We assume that a sheet of paper can be authenticated. Completeness of Sub-protocol 1.2 is self-evident. Secrecy of the shares follows from the properties of an n-VCX. If a printer attempts to read the document by activating the ink it will be evident following from the assumptions of the invisible ink. If a printer attempts to replace a valid document with a fake it will be evident following the assumptions regarding document authentication. Soundness is probabilistic and follows the cut-and-choose proof. If a printer prints nothing in a sub-pixel where it was to print invisible ink, it will either be covered by invisible ink from another share, and does not alter the intended outcome, or, it will not be covered by another share in which case it will be detectable by the cut-and-choose and attributable by examining the electronic shares. If a printer prints invisible ink in a sub-pixel where it was to print nothing, it will be detected similarly but is not attributable.

It is important to note that nothing fundamentally prevents an adversary in physical possession a document from activating the ink and reading its contents. The severity of this threat will depend greatly on the use-case. For example if the document contains a *unique secret*, additional physical security measures are necessary to protect document secrecy. Alternatively if the document contains an *arbitrary secret* (e.g., a new password), it may suffice for the recipient of a tampered document to simply request it be invalidated and a new one be issued.

Conclusion. In this paper we introduced oblivious printing. We presented three protocols: a generic protocol for obliviously printing an encrypted plaintext, a protocol with improved contrast for obliviously printing a random message, and third protocol to generate and obliviously print a DSA/Elgamal keypair. We then proposed a contrast optimization based on the AND-ing invisible inks and provided some example applications for electronic voting and digital cash.

Acknowledgements. We thank Jeremy Clark, Ian Goldberg, and Doug Stinson for helpful feedback. The authors are supported in part by NSERC; the first through a Canada Graduate Scholarship, the second through a Discovery Grant.

References

1. B. Adida, O. d. Marneffe, O. Pereira, and J.-J. Quisquater. Electing a university president using open-audit voting: Analysis of real-world use of Helios. In *EVT/WOTE*, 2009.
2. G. Ateniese, C. Blundo, A. D. Santis, and D. R. Stinson. Visual cryptography for general access structures. *Information and Computation*, 129:86–106, 1996.
3. J. D. R. Buchanan, R. P. Cowburn, A.-V. Jausovec, D. Petit, P. Seem, G. Xiong, D. Atkinson, K. Fenton, D. A. Allwood, and M. T. Bryan. Fingerprinting documents and packaging. *Nature*, 436:475, 2005.
4. R. T. Carback, D. Chaum, J. Clark, J. Conway, A. Essex, P. S. Hernson, T. Mayberry, S. Popoveniuc, R. L. Rivest, E. Shen, A. T. Sherman, and P. L. Vora. Scantegrity II municipal election at Takoma Park: the first E2E binding governmental election with ballot privacy. In *USENIX Security Symposium*, 2010.

5. D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. A. Ryan, E. Shen, and A. T. Sherman. Scantegrity II: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *EVT*, 2008.
6. D. Chaum, P. Y. A. Ryan, and S. Schneider. A practical voter-verifiable election scheme. In *ESORICS*, 2005.
7. J. Clark and U. Hengartner. Selections: Internet voting with over-the-shoulder coercion-resistance. In *FC*, 2011.
8. W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J. A. Halderman, and E. W. Felten. Fingerprinting blank paper using commodity scanners. In *IEEE Symposium on Security and Privacy*, 2009.
9. R. Cramer and I. Damgård. Multiparty computation from threshold homomorphic encryption. In *EUROCRYPT*, 2001.
10. A. Essex, J. Clark, U. Hengartner, and C. Adams. How to print a secret. In *HotSec*, 2009.
11. A. Essex, C. Henrich, and U. Hengartner. Single layer optical-scan voting with fully distributed trust. In *VOTE-ID*, 2011.
12. A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.
13. M. Jakobsson and A. Juels. Mix and match: Secure function evaluation via ciphertexts. In *ASIACRYPT*, 2000.
14. M. Jakobsson, A. Juels, and R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *USENIX Security Symposium*, pages 339–353, 2002.
15. A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *ACM WPES*, 2005.
16. O. Kafri and E. Keren. Encryption of pictures and shapes by random grids. *Optics Letters*, 12:6:377–379, 1987.
17. S.-S. Lee, J.-C. Na, S.-W. Sohn, C. Park, D.-H. Seo, and S.-J. Kim. Visual cryptography based on an interferometric encryption technique. *ETRI*, 24(5):373–380, 2002.
18. M. Naor and A. Shamir. Visual cryptography. In *EUROCRYPT*, 94.
19. C. A. Neff. Practical high certainty intent verification for encrypted votes. Technical report, VoteHere Whitepaper, 2004.
20. T. P. Pedersen. A threshold cryptosystem without a trusted party. In *EUROCRYPT*, 1991.
21. S. Popoveniuc and B. Hosp. An introduction to punchscan. In *WOTE*, 2006.
22. P. Y. A. Ryan and V. Teague. Pretty good democracy. In *Workshop on Security Protocols*, 2009.
23. K. Sako and J. Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *EUROCRYPT*, pages 393–403, 1995.
24. A. Sharma, L. Subramanian, and E. Brewer. Paperspeckle: Microscopic fingerprinting of paper. In *CCS*, 2011.
25. P. Tuyls, H. D. L. Hollmann, J. H. v. Lint, and L. Tolhuizen. Xor-based visual cryptography schemes. *Designs Codes and Cryptography*, 37:169–186, 2005.
26. United States Election Assistance Commission. 2008 election administration & voting survey report, 2008.
27. D. Q. Viet and K. Kurosawa. Almost ideal contrast visual cryptography with reversing. In *CT-RSA*, volume 2964, pages 353–365, 2004.
28. C.-N. Yang, editor. *Visual Cryptography and Secret Image Sharing*. CRC Press, 2011.