Spreading Alerts Quietly and the Subgroup Escape Problem[*]

James Aspnes[1][**], Zoë Diamadi[2], Kristian Gjøsteen[3][***],
René Peralta[4], and Aleksandr Yampolskiy[5][†]

[1] Yale University, Department of Computer Science,
New Haven, CT, USA
aspnes@cs.yale.edu

[2] LinkedIn, Mountain View, CA, USA
zdiamadi@linkedin.com

[3] Norwegian University of Science and Technology,
Department of Mathematical Sciences, 7491 Trondheim, Norway
kristian.gjosteen@math.ntnu.no

[4] Computer Security Division,
National Institute of Standards and Technology, USA
peralta@nist.gov

[5] SecurityScorecard Inc.
alex@securityscorecard.io

Abstract. We introduce a new cryptographic primitive called a **blind coupon mechanism** (BCM). In effect, a BCM is an authenticated bit commitment scheme, which is AND-homomorphic. We show that a BCM has natural and important applications. In particular, we use it to construct a mechanism for transmitting alerts undetectably in a message-passing system of n nodes. Our algorithms allow an alert to quickly propagate to all nodes without its source or existence being detected by an adversary, who controls all message traffic. Our proofs of security are based on a new **subgroup escape problem**, which seems hard on certain groups with bilinear pairings and on elliptic curves over the ring $\mathbb{Z}_n$.

Key words. Blind Coupon Mechanism, AND-Homomorphic Bit Commitment, Subgroup Es-

cape Problem, Elliptic Curves Over Composite Moduli, Anonymous Communication, Intrusion Detection.

1   Introduction

Motivation. As more computers become interconnected, chances increase greatly that an attacker may attempt to compromise system and network resources. It has become common to defend the network by running an Intrusion Detection System (IDS) on several of the network nodes, which we call sentinels. These sentinel nodes continuously monitor their local network traffic for suspicious activity. When a sentinel node detects an attacker's presence, it may want to alert all other network nodes to the threat. However, issuing an alert out in the open may scare the attacker away too soon and preclude the system administrator from gathering more information about the attacker's rogue exploits. Instead, we would like to propagate the alert without revealing the ids of the sentinel nodes or the fact that the alert is being spread.

We consider a powerful (yet computationally bounded) attacker who observes all message traffic and is capable of reading, replacing, and delaying circulating messages. Our work provides a cryptographic mechanism that allows an alert to spread through a population of network nodes at the full speed of an epidemic, while remaining undetectable to the attacker. As the alert percolates across the network, all nodes unwittingly come to possess the signal, making it especially difficult to identify the originator even if the secret key is compromised and the attacker can inspect the nodes' final states.

A New Tool: A Blind Coupon Mechanism. The core of our algorithms is a new cryptographic primitive called a **blind coupon mechanism** (BCM). The BCM is related to, yet quite different from, the notion of bit commitment. It consists of a set D_{SK} of **dummy coupons** and a set S_{SK} of **signal coupons** ($D_{SK} \cap S_{SK} = \emptyset$). The owner of the secret key SK can efficiently sample these sets and distinguish between their elements. We call the set of dummy and signal coupons, $D_{SK} \cup S_{SK}$, the set of **valid** coupons.

The BCM comes equipped with a **verification algorithm** $\mathcal{V}_{PK}(x)$ that checks whether x is indeed a valid coupon. There is also a probabilistic **combining algorithm** $\mathcal{C}_{PK}(x, y)$, that takes as input two valid coupons x, y and outputs a new coupon which is, with high probability, a signal coupon if and only if at least one of the inputs is a signal coupon. As suggested by the notation, both algorithms can be computed by anyone who has access to the public key PK of the blind coupon mechanism.
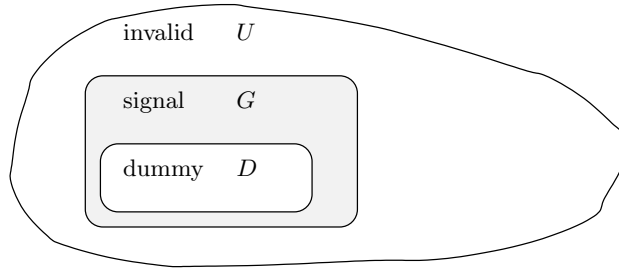
Fig. 1. Abstract group structure used in our BCM construction.

We regard the BCM secure if an observer who lacks the secret key SK (a) cannot distinguish between dummy and signal coupons (**indistinguishability**); (b) cannot engineer a signal coupon unless he is given another signal coupon as input (**unforgeability**).

OUR MAIN CONSTRUCTION. Our BCM construction uses an abstract group structure (U, G, D). Here, U is a finite set, $G \subseteq U$ is a cyclic group, and D is a proper subgroup of G. The elements of D will represent dummy coupons and the elements of $G \setminus D$ will be signal coupons (see Figure 1). More precisely, the group G is defined by the pair $(\mathcal{C}_{PK}(x, y), \mathcal{V}_{PK}(x))$. These functions can be thought of as given in the form of efficient algorithms or of oracles. The subgroup D is defined by an element $x \in D$ which generates D. Without loss of generality, we can assume U is the set of binary strings of length no more than a specified parameter n.

In order for the BCM to be secure, the following two problems must be hard on this group structure:

- **Subgroup Membership Problem**: Given generators for G and D and a random element $y \in G$, decide whether $y \in D$ or $y \in G \setminus D$.
- **Subgroup Escape Problem**: Given the set U, the group G and a generator of D, find an element of $G \setminus D$.

The subgroup membership problem has appeared in many different forms in the literature [12, 17, 19, 32, 33, 35, 37]. The subgroup escape problem has not been studied before. To provide more confidence in its validity, section 6 analyzes the problem in the generic group model.

Notice that the task of distinguishing a signal coupon from a dummy coupon (indistinguishability) and the task of forging a signal coupon (unforgeability) are essentially the subgroup membership and subgroup escape problems, respectively. The challenge thus becomes to find a concrete group structure (U, G, D) for which the subgroup membership and the subgroup escape problems are hard. We provide two instantiations of the group structure: one using groups with bilinear pairings, and one using elliptic curves over composite moduli.

WHY IS A BCM USEFUL? If signal coupons are used to encode a "0" and dummy coupons a "1", then a BCM can be viewed as an AND-homomorphic bit commitment scheme. It is indeed **hiding** because dummy and signal coupons appear the same to an outside observer. It is also **binding** because the sets of dummy and signal coupons are disjoint. We note that AND-homomorphic bit commitments were independently discovered by Boneh et al [7].

The BCM has various applications in and of itself. For example, it may prove useful for constructing zero-knowledge proofs of circuit satisfiability [9]. Returning to our original motivation, it can also be used to propagate alerts quickly and quietly throughout the network. We describe how this can be done below.

SPREADING ALERTS WITH THE BCM. During the initial network setup, the network administrator generates the BCM's public and secret keys. He then distributes signal coupons to sentinel nodes. All other nodes receive dummy coupons. In our mechanism, nodes continuously transmit either dummy or signal coupons with all nodes initially transmitting dummy coupons. Sentinel nodes switch to sending signal coupons when they detect the attacker's presence. The BCM's combining algorithm allows dummy and signal coupons to be combined so that a node can propagate signal coupons without having to know that it has received any, and so that an attacker (who can observe all message traffic) cannot detect where or when signals are being transmitted within the stream of dummy messages.

In addition, the BCM's verification algorithm defends against Byzantine nodes [27]: While Byzantine nodes can replay old dummy messages instead of relaying signals, they cannot flood the network with invalid coupons, thereby preventing an alert from spreading; at worst, they can only act like crashed nodes.

We prove that if the underlying BCM is secure, then the attacker cannot distinguish between executions where an alert was sent and executions where no alert was sent. The time to spread the alert to all nodes will be determined by the communications model and alert propagation strategy. At any point in time, the network administrator can sample the state of some network node and check if it possesses a signal coupon.

PAPER ORGANIZATION. The rest of the paper is organized as follows. We begin by covering some technical preliminaries in Section 2. Then in Section 3, we formally define the notion of a blind coupon mechanism and sketch an abstract group structure, which will allow us to implement it. In Section 4, we provide two concrete instantiations of this group structure using certain bilinear groups and elliptic curves over the ring $\mathbb{Z}_n$. In Section 5, we show how the BCM can be used to spread alerts quietly throughout a network. In Section 6, we analyze the hardness of the subgroup

escape problem in the generic group model. Some related work is discussed in Section 7. Conclusions and open problems appear in Section 8.

2 Preliminaries

This section reviews our notation and some basic facts about elliptic curves, which form the cornerstone of our constructions.

2.1 Notation

In this paper, we use standard notation from [6].

Let $A(\cdot)$ be an algorithm. Then $y \leftarrow A(x)$ denotes that y was obtained by running A on input x. In case A is deterministic, this y is unique. If A is probabilistic, then $A(x)$ describes a probability and $y \leftarrow A(x)$ denotes sampling y from the probability space. If U is a probability space, then $y \xleftarrow{\$} U$ denotes that y was sampled from U. If S is a finite set, then $y \xleftarrow{\$} S$ denotes that y was chosen from S uniformly at random.

Let b be a Boolean predicate. The notation $[b(y) \mid y \leftarrow A(x)]$ denotes the event that $b(y)$ is true after y is output by A on input x. Similarly, the statement

$$\Pr\left[b(x_0, \ldots, x_n) \mid x_0 \xleftarrow{\$} S, x_1 \leftarrow A_1(x_0), \ldots, x_n \leftarrow A_n(x_0, \ldots, x_{n-1}) \right]$$

denotes the probability that the predicate $b(x_0, \ldots, x_n)$ will be true after the sequential execution of algorithms A_i. Here, x_0 is drawn uniformly from set S and each subsequent x_i is drawn from a distribution on algorithm A_i's output, possibly depending on previous inputs.

The **statistical distance** between a pair of random variables A, B is defined to be $\mathrm{Dist}(A, B) \stackrel{def}{=} \frac{1}{2} \sum_x |\Pr[A = x] - \Pr[B = x]|$ is

Finally, we will say that $negl(k) : \mathbb{N} \mapsto (0, 1)$ is a **negligible function** if for every $c > 0$, for all sufficiently large k, $negl(k) < 1/k^c$

2.2 Elliptic Curves

Let $p \geq 5$ be a prime. Consider the set $\bar{U}_p = \mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p \setminus \{(0, 0, 0)\}$. Let $\sim$ be the equivalence relation on $\bar{U}_p$ such that $(x, y, z) \sim (x', y', z')$ if and only if there exists $\lambda \in \mathbb{Z}_p^*$ such that $(x, y, z) = (\lambda x', \lambda y', \lambda z')$. Let U_p be the set of equivalence classes in $\bar{U}_p$. We denote the equivalence class of (x, y, z) as $(x : y : z)$.

An elliptic curve over $\mathbb{Z}_p$ is defined by the equation

$$E : Y^2 Z = X^3 + aXZ^2 + bZ^3,$$

where $a, b \in \mathbb{Z}_p$ are such that $4a^3 + 27b^2 \neq 0$. The set of points on $E/\mathbb{Z}_p$ is the set of equivalence classes $(x : y : z) \in U$ satisfying $y^2 z = x^3 + axz^2 + bz^3$, and is denoted by $E(\mathbb{Z}_p)$. We get an abelian group structure on $E(\mathbb{Z}_p)$ by taking the point $(0 : 1 : 0)$ as the identity element and postulating that any three colinear points sum to zero [43].

Usually in cryptography, elliptic curves with (close to) a prime number of points are required. Such curves can be found by counting points on random elliptic curves until a suitable curve is found, since there are many suitable curves and point counting is fast [41]. When curves with a prescribed number of points are required, the so-called complex multiplication techniques can be used [28] to find a prime and a suitable curve.

There are several non-degenerate bilinear pairings on elliptic curves. For some families of elliptic curves, these can be computed efficiently [31]. The first application of these pairings to cryptography was the so-called MOV attack [30] where the pairing was used to reduce the elliptic curve discrete logarithm problem to the finite field discrete logarithm problem, which is much easier. Constructive applications of pairings first appeared in [24].

Let n be an integer greater than 1 and not divisible by 2 or 3. We first introduce projective coordinates over $\mathbb{Z}_n$. Consider the set $\bar{U}_n$ of triples $(x, y, z) \in \mathbb{Z}_n^3$ such that the ideal generated by x, y and z is $\mathbb{Z}_n$. Let $\sim$ be the equivalence relation on $\bar{U}_n$ defined by $(x, y, z) \sim (x', y', z')$ if and only if there exists $\lambda \in \mathbb{Z}_n^*$ such that $(x, y, z) = (\lambda x', \lambda y', \lambda z')$. Let U_n be the set of equivalence classes in $\bar{U}_n$. We denote the equivalence class of (x, y, z) as $(x : y : z)$.

An elliptic curve over $\mathbb{Z}_n$ is defined by the equation

$$E : Y^2 Z \equiv X^3 + aXZ^2 + bZ^3 \pmod{n},$$

where a, b are integers satisfying $\gcd(4a^3 + 27b^2, n) = 1$. The set of points on $E/\mathbb{Z}_n$ is the set of equivalence classes $(x : y : z) \in U_n$ satisfying $y^2 z \equiv x^3 + axz^2 + bz^3 \pmod{n}$, and is denoted by $E(\mathbb{Z}_n)$. Note that if n is prime, these definitions correspond to the usual definitions for projective coordinates over prime fields.

Let p and q be primes, and let $n = pq$. Let $E_p : Y^2 Z = X^3 + a_p X Z^2 + b_p Z^3$ and $E_q : Y^2 Z = X^3 + a_q X Z^2 + b_q Z^3$ be elliptic curves defined over $\mathbb{F}_p$ and $\mathbb{F}_q$, respectively. We can use the Chinese remainder theorem to find a and b yielding an elliptic curve $E : Y^2 Z = X^3 + aXZ^2 + bZ^3$ over $\mathbb{Z}_n$ such that the reduction of E modulo p gives E_p and likewise for q.

The Chinese remainder theorem gives a bijection

$$E(\mathbb{Z}_n) \xrightarrow{\sim} E_p(\mathbb{F}_p) \times E_q(\mathbb{F}_q)$$

which in turn induces a group operation on $E(\mathbb{Z}_n)$. For almost all points in $E(\mathbb{Z}_n)$, the usual group operation formulae for the finite field case will compute the induced group operation. When they fail, the attempted operation gives a factorization of the composite modulus n. Unless $E_p(\mathbb{F}_p)$ or $E_q(\mathbb{F}_q)$

has smooth or easily guessable order, this will happen only with negligible probability (see [15] for more details).

3 Blind Coupon Mechanism

The critical component of our algorithms that allows information to propagate undetectably among the network nodes is a cryptographic primitive called a **blind coupon mechanism** (BCM). In Section 3.1, we give a formal definition of the BCM and its security properties. In Section 3.2, we describe an abstract group structure that will allow us to construct a BCM.

3.1 Definitions

Definition 1. *A **blind coupon mechanism** is a tuple of PPT algorithms* $(\mathcal{G}, \mathcal{V}, \mathcal{C}, \mathcal{D})$ *in which:*

- $\mathcal{G}(1^k)$ *is a probabilistic **key generation algorithm** that outputs* (PK, SK, d, s) *such that*
  - *the **public key** PK defines a universe set U_{PK} and a set of **valid coupons** $G_{PK} \subseteq U_{PK}$;*
  - *the **secret key** SK defines sets $D_{SK}, S_{SK} \subseteq U_{PK}$ of **dummy coupons** and **signal coupons**, respectively;*
  - *membership in U_{PK} is efficiently decidable given PK;*
  - $d \in D_{SK}$ *and* $s \in S_{SK}$;
  - $D_{SK} \cap S_{SK} = \emptyset$, *and* $D_{SK} \cup S_{SK} = G_{PK}$.
- $\mathcal{V}_{PK}(y)$, *the deterministic **verification algorithm**, takes as input a coupon y and returns 1 if y is valid and 0 if it is invalid.*
- $\mathcal{C}_{PK}(x, y)$, *the probabilistic **combining algorithm**, takes as input two valid coupons x, y and produces a new coupon. Let U_D be the uniform distribution on D_{SK} and U_S be the uniform distribution on S_{SK}. Then for any pair of keys (PK, SK) output by $\mathcal{G}(1^k)$ and valid coupons x, y, $\mathcal{C}$ satisfies*

$$
\begin{cases}
\mathrm{Dist}(\mathcal{C}_{PK}(x, y), U_D) = negl(k) & \text{if } x, y \in D_{SK}, \\
\mathrm{Dist}(\mathcal{C}_{PK}(x, y), U_S) = negl(k) & \text{otherwise.}
\end{cases}
$$

  This gives $\mathcal{C}$ the properties listed in Fig. 2.
- $\mathcal{D}_{SK}(y)$, *the deterministic **decoding algorithm**, takes as input a valid coupon y. It returns 0 if y is a dummy coupon and 1 if y is a signal coupon. (Note that $\mathcal{D}_{SK}(y)$ is undefined if y is an invalid coupon. Hence $\mathcal{V}_{PK}$ should be used to validate the coupon before it is decoded.)*

The BCM may be established either by an external trusted party or jointly by the application participants, running a distributed key generation protocol (*e.g.*, one could use a variant of [2]). In this paper, we assume a trusted dealer (the network administrator) who runs the key generation

x	y	$\mathcal{C}_{PK}(x,y)$
D_{SK}	D_{SK}	D_{SK}
D_{SK}	S_{SK}	S_{SK}
S_{SK}	D_{SK}	S_{SK}
S_{SK}	S_{SK}	S_{SK}

Fig. 2. Properties of the combining algorithm.

algorithm and distributes signal coupons to sentinel nodes at the start of the system execution. In a typical algorithm, the nodes will continuously exchange coupons with each other. The combining algorithm $\mathcal{C}_{PK}$ enables nodes to locally and efficiently combine their coupons with coupons of other nodes. The verification function $\mathcal{V}_{PK}$ is used to prevent the propagation of invalid coupons. Thus, an adversary cannot slow down or halt the propagation of an alert signal by flooding the system with invalid coupons.

For this application, we require the BCM to have certain specific security properties.

Definition 2. *We say that a blind coupon mechanism* $(\mathcal{G}, \mathcal{V}, \mathcal{C}, \mathcal{D})$ *is* **secure** *if it satisfies the following requirements:*

1. **Indistinguishability**: *Given a valid coupon* y*, the adversary cannot tell whether it is a signal or a dummy coupon with probability better than* $1/2$*. Formally, for any PPT algorithm* $\mathcal{A}$*,*

$$\left| \Pr\left[b = b' \middle| \begin{array}{c} (PK, SK, d, s) \leftarrow \mathcal{G}(1^k); \\ y_0 \overset{\$}{\leftarrow} D_{SK}; y_1 \overset{\$}{\leftarrow} S_{SK}; \\ b \overset{\$}{\leftarrow} \{0,1\}; b' \leftarrow \mathcal{A}(1^k, PK, d, y_b) \end{array} \right] - \frac{1}{2} \right| \leq negl(k)$$

2. **Unforgeability**: *The adversary is unlikely to fabricate a signal coupon without the use of another signal coupon as input[6]. Formally, for any PPT algorithm* $\mathcal{A}$*,*

$$\Pr\left[y \in S_{SK} \middle| \begin{array}{c} (PK, SK, d, s) \leftarrow \mathcal{G}(1^k); \\ y \leftarrow \mathcal{A}\left(1^k, PK, d\right) \end{array} \right] \leq negl(k)$$

To build the reader's intuition, we describe a straw-man construction of a BCM. Suppose we are given a semantically secure encryption scheme $\mathcal{E}(\cdot)$ and a set-homomorphic signature scheme $\mathrm{SIG}(\cdot)$ (see, for example Johnson *et al.* [23]). This signature scheme allows anyone possessing sets $x, y \subseteq \mathbb{Z}_p$ and their signatures $\mathrm{SIG}(x), \mathrm{SIG}(y)$ to compute $\mathrm{SIG}(x \cup y)$ and $\mathrm{SIG}(w)$ for any $w \subseteq x$. We represent dummy coupons by a variable-length vector of encrypted zeroes; *e.g.*, $x = (\mathcal{E}(0), \ldots, \mathcal{E}(0))$. The

---

[6] The adversary, however, can easily generate polynomially many dummy coupons by using $\mathcal{C}_{PK}(\cdot, \cdot)$ with the initial dummy coupon d that he receives.

signal coupons are represented by a vector of encryptions that contains at least one encryption of a non-zero element; *e.g.*, $y = (\mathcal{E}(0), \ldots, \mathcal{E}(0), \mathcal{E}(1))$. To prevent the adversary from forging coupons, the coupons are signed with the set-homomorphic signature. The combining operation is simply the set union: $\mathcal{C}_{PK}\ (x, \mathrm{SIG}(x)), (y, \mathrm{SIG}(y))\ =\ x \cup y, \mathrm{SIG}(x \cup y)\ $. The drawback of this construction is immediate: as coupons are combined and passed around the network, they quickly grow very large. Constructing a BCM with no expansion of coupons is more challenging. We describe such a construction next.

3.2 Abstract Group Structure

We sketch the abstract group structure that will allow us to implement a secure and efficient BCM. Concrete instantiations of this group structure are provided in Section 4.

Let $\Gamma = \{\Gamma_k\}$ be a family of sets of tuples (U, G, D, d, s), and let $\mathcal{G}'$ be a PPT algorithm that on input 1^k samples from Γ_k according to some distribution. For the tuple (U, G, D, d, s), U is a finite set with an efficient membership test, and G is a subset of U. G has a group structure: it is a cyclic group generated by s. D is a subgroup of G generated by d, such that the factor group G/D has prime order $|G|/|D|$. The orders of D and G/D are bounded by 2^k (hence, $|G|$ is bounded by 2^{2k}). Moreover, $|G|/|U| \leq negl(k)$ and $|D|/|G| \leq negl(k)$. Henceforth, we assume that these groups and their elements have some concise description, which can be passed as an argument to our algorithms. We note that given a generator of a finite cyclic group and a reasonable upper bound on the size of the group, one can always sample arbitrarily close to uniformly from the group.

Suppose there exist an efficient, deterministic algorithm for distinguishing elements of G from elements of $U \setminus G$ and an efficient algorithm for computing the group operation in G. Then our BCM $(\mathcal{G}, \mathcal{V}, \mathcal{C}, \mathcal{D})$ is as follows.

- The **key generation algorithm** $\mathcal{G}(1^k)$ runs $\mathcal{G}'$ to sample (U, G, D, d, s) from Γ_k, and outputs the public key $PK = (U, G, d, k)$, the secret key $SK = |D|$, as well as d and s.
  The elements of D will represent dummy coupons, the elements of $G \setminus D$ will represent signal coupons, and the elements of $U \setminus G$ will be invalid coupons (see Figure 1).
- The **verification algorithm** $\mathcal{V}_{PK}(y)$ checks that the coupon y is in G.
- The **combining algorithm** $\mathcal{C}_{PK}(x, y)$ is simply the group operation combined with randomization. For input $x, y \in G$, sample r_0, r_1 and r_2 uniformly at random from $\{0, 1, \ldots, 2^{2k} - 1\}$, and output $r_0 d + r_1 x + r_2 y$.
- Because $|D| \cdot y = 0$ if and only if $y \in D$, the **decoding algorithm** $\mathcal{D}_{SK}$ outputs 0 if $|D| \cdot y = 0$, otherwise 1.

Theorem 1. *The above construction $(\mathcal{G}, \mathcal{V}, \mathcal{C}, \mathcal{D})$ is a BCM.*

Proof. All that is left to prove is that the combining algorithm has the required properties. Fix k and (U, G, D, d, s) sampled from Γ_k. We need to show that on input two coupons $x, y \in G$, the distribution of the output $r_0 d + r_1 x + r_2 y$ is close to uniform in D if $x, y \in D$, and close to uniform in $G \setminus D$ otherwise.

We start by assuming that the group orders $|G|$ and $|D|$ are known, and sample r_0 from the uniform distribution on $\{0, 1, \ldots, |D|-1\}$ and r_1, r_2 from the uniform distribution on $\{0, 1, \ldots, |G/D|-1\}$. Since d is a generator for D, $r_0 d$ will be uniformly distributed in D. Therefore, if $x, y \in D$, the sum $r_0 d + r_1 x + r_2 y$ will also be uniformly distributed in D.

If $x \notin D$, then the residue class $x + D$ is non-zero, hence the class is a generator for the prime-order factor group G/D. This means that $r_1(x + D)$ is uniformly distributed in G/D. If $r_1(x + D)$ is uniformly distributed in G/D, and $r_0 d$ is uniformly distributed in D, the sum $r_0 d + r_1 x$ is uniformly distributed in G, and therefore $r_0 d + r_1 x + r_2 y$ will also be uniformly distributed in G. The same argument applies if $y \notin D$.

Now we do away with the assumption that $|G|$ and $|D|$ are known. Note that for any generator z of a group of order at most 2^k, if a is sampled from the uniform distribution on $\{0, 1, \ldots, 2^{2k}-1\}$, the distribution of az is 2^{-k}-close to uniform over the group. Hence, when r_0, r_1, r_2 are sampled uniformly from $\{0, 1, \ldots, 2^{2k}-1\}$, the distribution of the output $r_0 d + r_1 x + r_2 y$ will be $3 \cdot 2^{-k}$-close to uniform on D when $x, y \in D$ or on G when one of x or y is not in D. In the latter case, we note that the uniform distribution on G is $|D|/|G|$-close to the uniform distribution on $G \setminus D$, which is negligible in k. $\qquad\qquad\square$

The indistinguishability and unforgeability properties of the BCM will depend on the hardness assumptions described below.

Definition 3. *The **subgroup membership problem** for $(\Gamma, \mathcal{G}')$ asks: given a tuple (U, G, D, d, s) sampled from Γ using $\mathcal{G}'$ and $y \in G$, decide whether $y \in D$ or $y \in G \setminus D$.*

The subgroup membership problem is hard if for any PPT algorithm $\mathcal{A}$,

$$\Pr\left[b' = b \,\middle|\, \begin{array}{c} (U, G, D, d, s) \leftarrow \mathcal{G}'(1^k); \\ y_0 \xleftarrow{\$} D; y_1 \xleftarrow{\$} G \setminus D; \\ b \xleftarrow{\$} \{0, 1\}; b' \leftarrow \mathcal{A}(U, G, D, d, s, y_b) \end{array} \right] - \frac{1}{2} \; \leq negl(k).$$

Various subgroup membership problems have been extensively studied in the literature, and examples include the Decision Diffie-Hellman problem [12], the quadratic residue problem [19], among others [32, 35, 37]. Our constructions, however, are more closely related to the problems described in [17, 33].

Definition 4. *The **subgroup escape problem** for $(\Gamma, \mathcal{G})$ asks: given U, G, D and the generator d for D from the tuple (U, G, D, d, s) sampled from Γ using $\mathcal{G}'$, find an element $y \in G \setminus D$.*

The subgroup escape problem is hard if for any PPT algorithm $\mathcal{A}$,

$$\Pr\left[y \in G \setminus D \;\middle|\; \begin{array}{c}(U, G, D, d, s) \leftarrow \mathcal{G}'(1^k); \\ y \leftarrow \mathcal{A}(U, G, D, d)\end{array}\right] \leq negl(k).$$

The subgroup escape problem has, to our knowledge, not appeared in the literature before. It is clear that unless $|G|/|U|$ is negligible, finding elements of $G \setminus D$ cannot be hard. We show in Section 6 that if $|G|/|U|$ is negligible, the subgroup escape problem is provably hard in the generic model. We also note that the problem of generating a signal coupon from polynomially many dummy coupons is essentially the subgroup escape problem.

We now prove that our BCM construction on (U, G, D) is secure.

Theorem 2. *Let Γ be as above. If the subgroup membership problem and the subgroup escape problem for Γ are hard, then the corresponding BCM is secure.*

Proof. Fix k and (U, G, D, d, s) sampled from Γ_k.

We first prove the indistinguishability property. Let $\mathcal{A}$ be an adversary against indistinguishability, taking a public key $PK = (U, G, d, k)$ and a coupon $y \in G$ as input, and outputting a bit b'. We construct an adversary $\mathcal{B}$ against the subgroup membership problem taking the problem instance (U, G, D, d, s) and $y \in G$ as input and outputting a bit. $\mathcal{B}$ runs $\mathcal{A}$ with input $PK = (U, G, d, k)$ and y, and simply outputs $\mathcal{A}$'s output bit.

Since the BCM key generation algorithm essentially samples an instance of the subgroup membership problem, the input to $\mathcal{A}$ will be correctly distributed. Also, if $\mathcal{A}$ correctly identifies a signal or dummy coupon, $\mathcal{B}$ will correctly decide if the element is in the subgroup or not. Therefore, if the subgroup membership problem is hard, $\mathcal{B}$ must have negligible advantage, hence $\mathcal{A}$ must also have negligible advantage and indistinguishability will be satisfied.

Next, we deal with forging. Let $\mathcal{A}$ be an adversary against unforgeability, taking a public key $PK = (U, G, d, k)$ as input and outputting a coupon y. We construct an adversary $\mathcal{B}$ against the subgroup escape problem taking the problem instance (U, G, D, d) as input and outputting an element $y \in G$. $\mathcal{B}$ runs $\mathcal{A}$ with input $PK = (U, G, d, k)$ and simply outputs $\mathcal{A}$'s coupon.

Again, since the BCM key generation algorithm essentially samples an instance of the subgroup escape problem, the input to $\mathcal{A}$ will be correctly distributed. If $\mathcal{A}$ succeeds in finding a valid signal coupon, this will be an element in $G \setminus D$, hence $\mathcal{B}$ will succeed in escaping from the subgroup. We conclude that if the subgroup escape problem is hard, $\mathcal{A}$ must have negligible advantage in forging valid signal coupons.

This concludes the proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

4 Constructing the BCM

We now give two instantiations of the abstract group structure (U, G, D) described in the previous section. In Section 4.1, we describe one such construction on elliptic curves over a composite modulus. In Section 4.2, we describe an alternate BCM construction on groups equipped with bilinear pairings. These constructions can be used to undetectably transmit a one-shot signal throughout the network. Finally, in Section 4.3, we describe how the BCM's bandwidth can be further expanded.

4.1 BCM on Elliptic Curves Modulo Composites

Let $p, q, \ell_1, \ell_2, \ell_3$ be primes, and suppose we have elliptic curves $E_p/\mathbb{F}_p$ and $E_q/\mathbb{F}_q$ such that $\#E_p(\mathbb{F}_p) = \ell_1\ell_2$ and $\#E_q(\mathbb{F}_q) = \ell_3$. Curves of this form can be found using complex multiplication techniques [5, 28] by first finding ℓ_1, ℓ_2, ℓ_3, then finding appropriate primes and curves. Alternatively, ℓ_3 and q could be found by counting points [41] on random curves, avoiding one instance of complex multiplication curves.

With $n = pq$, we can find $E/\mathbb{Z}_n$ such that $\#E(\mathbb{Z}_n) = \ell_1\ell_2\ell_3$. Let U be the projective plane modulo n, let G be $E(\mathbb{Z}_n)$, and let D be the subgroup of order $\ell_1\ell_3$. The public key is $PK = (G, D, n)$, while the secret key is $SK = (p, q, \ell_1, \ell_2, \ell_3)$. To describe the groups G and D, we publish the elliptic curve equation and the generator for D. This gives away enough information to perform group operations in G, check membership in G, and generate new elements in D (but not in G).

VERIFICATION FUNCTION For any equivalence class $(x : y : z)$ in U, it is easy to decide if $(x : y : z)$ is in $E(\mathbb{Z}_n)$ or not, simply by checking if $y^2 z \equiv x^3 + axz^2 + bz^3 \pmod{n}$.

SUBGROUP MEMBERSHIP PROBLEM For the curve $E_p(\mathbb{F}_p)$, distinguishing the elements of prime order from the elements of composite order seems to be hard, unless it is possible to factor the group order [17].

Counting the number of points on an elliptic curve defined over a composite number is equivalent to factoring the number [26, 29]. Therefore, the group order $E_p(\mathbb{F}_p)$ is hidden.

When the group order is hidden, it cannot be factored. It therefore seems likely that the subgroup of $E(\mathbb{Z}_n)$ of order $\ell_1\ell_3$ is hard to distinguish from the rest of the points on the curve, as long as the integer n is hard to factor.

SUBGROUP ESCAPE PROBLEM Anyone capable of finding a random point on the curve will, with overwhelming probability, be able to find a point outside the subgroup D.

Finding a random point on an elliptic curve over a field is easy: Choose a random x-coordinate and solve the resulting quadratic equation. It has rational solutions with probability close to $1/2$.

This does not work for elliptic curves over the ring $\mathbb{Z}_n$, since solving square roots modulo n is equivalent to factoring n. One could instead try to choose a y-coordinate and solve for the x-coordinate, but solving cubic equations in $\mathbb{Z}_n$ seems no easier than finding square roots.

One could try to find x and y simultaneously, but there does not seem to be any obvious strategy. This is in contrast to quadratic curves, where Pollard [40] gave an algorithm to find solutions of a quadratic equation modulo a composite (which broke the Ong-Schnorr-Shamir signature system [36]). These techniques do not seem to apply to the elliptic curve case.

Finding a lift of the curve over the integers does not seem promising. While torsion points are fairly easy to find, they will not exist if the curve $E/\mathbb{Z}_n$ does not have points of order less than or equal to 12. If we allow $E/\mathbb{Z}_n$ to have points of small order that are easily found, we can simply include them in the subgroup D.

Finding rational non-torsion points on curves defined over $\mathbb{Q}$ is certainly non-trivial, and seems impossibly hard unless the point on the lifted curve has small height [44]. There does not seem to be any obvious way to find a lift with rational points of small height (even though they certainly exist).

What if we already know a set of points on the curve? If we are given $P_1, P_2, P_3 \in E(\mathbb{Z}_n)$, we can find, unless the points are collinear, a quadratic curve

$$C : YZ = \alpha X^2 + \beta XZ + \gamma Z^2$$

defined over $\mathbb{Z}_n$ that passes through P_1, P_2, P_3 and a fourth point P_4 which is easy to compute. Considering divisors, it is easy to show that the fourth intersection point P_4 is the inverse sum of the three known points.

If points of the curve only yield new points via the group operation, and it seems hard to otherwise find points on $E(\mathbb{Z}_n)$, it is reasonable to assume that $E(\mathbb{Z}_n)$ and its subgroup, as described above, yield a hard subgroup escape problem.

PARAMETER SIZES It seems difficult to distinguish the special curve E, and the special points on E disclosed in the system, from a curve chosen at random together with a point on the curve. This suggests that the elliptic curve E does not contribute any knowledge that would aid in factoring n. The most efficient attack on the Subgroup Membership Problem and the Subgroup Escape Problem therefore seems to be to factor the modulus n directly. The fact that one of the prime factors of n was selected at the same time as an elliptic curve does not seem to make n any easier to factor. Hence we can use the commonly accepted parameter sizes, where the two prime factors of n should be about the same size, and n should be between 1000 and 3000 bits in length depending on the required security level (see table 4 of [3]).

4.2   BCM on Groups With Bilinear Pairings

Let p, ℓ_1, ℓ_2, and ℓ_3 be primes such that $p + 1 = 6\ell_1\ell_2\ell_3$. Here, ℓ_1, ℓ_2, ℓ_3 must be distinct and larger than 3. The elliptic curve $E : Y^2 = X^3 + 1$ defined over $\mathbb{F}_p$ is supersingular and has order $p + 1$. Because $\mathbb{F}_{p^2}^*$ has order $p^2 - 1 = (p+1)(p-1)$, there is a modified Weil pairing $\hat{e} : E(\mathbb{F}_p) \times E(\mathbb{F}_p) \to \mathbb{F}_{p^2}^*$. This pairing is known to be bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in E(\mathbb{F}_p)$ and $a, b \in \mathbb{Z}_p$. It can be computed as described in [31]. There are also other constructions for such curves that are not supersingular, such as [8].

Let $U = E(\mathbb{F}_p)$, and let G and D be the subgroups of $E(\mathbb{F}_p)$ of orders $\ell_1\ell_2$ and ℓ_1, respectively. We also let P be a point in $E(\mathbb{F}_p)$ of order $6\ell_1\ell_2\ell_3$, and let R be a point of order $6\ell_3$ in $E(\mathbb{F}_p)$, say $R = \ell_1\ell_2 P$. The public key is $PK = (G, D, p, R)$ and the secret key is $SK = (\ell_1, \ell_2, \ell_3)$. The pairing $\hat{e}$ allow us to describe G in the public key without giving away secret information.

VERIFICATION FUNCTION We claim that for any point $Q \in E(\mathbb{F}_p)$, $Q \in G$ if and only if $\hat{e}(Q, R)$ is equal to 1. If $Q \in G$, then Q has order $\ell_1\ell_2$ and for some integer s, $Q = 6s\ell_3 P$. Then

$$\hat{e}(Q, R) = \hat{e}(6s\ell_3 P, \ell_1\ell_2 P) = \hat{e}(P, P)^{6s\ell_1\ell_2\ell_3} = 1.$$

So the point R and the pairing $\hat{e}$ allows us to determine if points are in G or in $U \setminus G$.

SUBGROUP MEMBERSHIP PROBLEM Distinguishing the subgroup D (the points of order ℓ_1) from G (the points of order $\ell_1\ell_2$) can easily be done if the integer $\ell_1\ell_2\ell_3$ can be factored. In general, factoring seems to be the best way to distinguish the various subgroups of $E(\mathbb{F}_p)$.

Because we do not reveal any points of order ℓ_2 or $\ell_2\ell_3$, it seems impossible to use the pairing to distinguish the subgroup D in this way. (Theorem 1 of [17] assumes free sampling of any subgroup, which is why it and the pairing cannot be used to distinguish the subgroups of $E(\mathbb{F}_p)$.) It therefore seems reasonable to assume that the subgroup membership problem for G and D is hard, which will provide indistinguishability.

SUBGROUP ESCAPE PROBLEM For a general cyclic group of order $\ell_1\ell_2\ell_3$, it is easy to find elements of order $\ell_1\ell_2$ if ℓ_3 is known. Unless ℓ_3 is known, it is hard to find elements of order $\ell_1\ell_2$, and knowing elements of order ℓ_1 does not help.

For our concrete situation, factoring the integer $\ell_1\ell_2\ell_3$ into primes seems to be the best method for solving the problem. If the primes ℓ_1, ℓ_2 and ℓ_3 are chosen carefully to make the product $\ell_1\ell_2\ell_3$ hard to factor, it seems reasonable to assume that the subgroup escape problem for U, G and D is hard.

PARAMETER SIZES It seems reasonable to assume that factoring $\ell_1\ell_2\ell_3$ to be the best method for attacking the Subgroup Membership Problem and the Subgroup Escape Problem. Computing

discrete logarithms on the elliptic curve or in the associated finite field will be significantly more difficult than factoring with current methods.

Since this number has three prime factors, we also need to consider Lenstra's integer factoring algorithm [29] in addition to the usual number field sieve. We know that a lower bound for Lenstra's method is approximately $\exp(\sqrt{2\log(\ell)\log\log\ell})$ point additions, where ℓ is about the size of the prime divisors. This suggest that for an 80-bit security level, the prime divisors must be about 400 bits each, leading to a 1200 bit n, somewhat larger than the 1000 bit n dictated by the number field sieve. For a 128 bit security level, we get prime divisors of about 900 bits, less than the 3000 bit n dictated by the number field sieve.

4.3   Extending the BCM's Bandwidth

The blind coupon mechanism allows to undetectably transmit a single bit. Although this is sufficient for our network alert application, sometimes we may want to transmit longer messages.

TRIVIAL CONSTRUCTION. By using multiple blind coupon schemes over different moduli in parallel, we can transmit longer messages. Each m-bit message $x = x_1 \ldots x_m$ is represented by a vector of coupons $\langle c_1, \ldots, c_{2m} \rangle$, where each c_i is drawn from a different scheme. Each node applies its algorithm in parallel to each of the entries in the vector, verifying each coupon independently and applying the appropriate combining operation to each c_i.

A complication is that an adversary given a vector of coupons might choose to propagate only some of the c_i, while replacing others with dummy coupons. In our model, this is impossible to prevent. But we allow the receiver to detect missing message bits by encoding each bit x_i in two coupons: for $x_i = 0$, we put a signal coupon in c_{2i-1} and a dummy coupon in c_{2i}, and reverse for $x_i = 1$. A signal coupon in either position tells the receiver both the value of the bit and that the receiver has successfully received it. Dummy coupons in both positions indicate a missing bit.

Alas, we must construct and run $\Omega(m)$ blind coupon schemes in parallel to transmit m bits.

BETTER CONSTRUCTION. Some additional improvements in efficiency are possible. As before, our group structure is (U, G, D). Suppose our cyclic group G has order $n_0 p_1 \cdots p_m$, where p_i are distinct primes. Let D be the subgroup of G of order n_0.

An m-bit message $x = x_1 \ldots x_m$ is encoded by a coupon $y \in G$, whose order is divisible by $\prod_{i \,:\, x_i=1} p_i$. For all i, we can find an element $g_i \in G$ of order $n_0 p_i$. We can thus let $y = g_1^{r_1 x_1} \cdots g_m^{r_m x_m}$ for random $r_1, \ldots, r_m \in \{0, 1, \ldots, 2^{2k} - 1\}$.

When we combine two coupons y_1 and y_2, it is possible that the order of their combination $\mathcal{C}_{PK}(y_1, y_2)$ is less than the l.c.m. of their respective orders. However, if the primes p_i are sufficiently large, this is unlikely to happen.

In Section 4.1, n_0 is a product of two moderately large primes, while the other primes can be around 2^{80}. For the construction from Section 4.2, n_0 is prime, but every prime must be fairly large to counter elliptic curve factorization. We note that strictly speaking, this changes the underlying problems, but we do not believe the changes are significant.

This technique allows us to transmit messages of quite restricted bandwidth. It remains an open problem whether some other tools can be used to achieve higher capacity without a linear blow-up in message size.

5   Spreading Alerts with the BCM

In this section, we show how the BCM can be used to spread an alert quietly and quickly throughout a network.

We consider a very general message-passing model in which each node P_i has a "split brain," consisting of an **update algorithm** $\mathcal{U}_i$ that is responsible for transmitting and combining coupons, and a **supervisor algorithm** $\mathcal{S}_i$ that may insert a signal coupon into the system at some point. The supervisor algorithm $\mathcal{S}_i$ of sentinel nodes initially hands out dummy coupons. When an attacker's presence is detected, the algorithm switches to sending signal coupons. On a non-sentinel node, $\mathcal{S}_i$ always doles out dummy coupons. Note that the supervisor algorithm always uses the combining algorithm when handing out coupons. The update algorithm $\mathcal{U}_i$ in each node may behave arbitrarily; the intent is that it represents an underlying strategy for spreading alerts whose actions do not depend on whether the node is transmitting dummy or signal coupons.

The nodes carry out these operations under the control of a PPT **attacker** $\mathcal{A}$ (who wants to remain undetectable) that can observe all the external operations of the nodes and may deliver any message to any node at any time, including messages of its own invention.

An **execution trace** in this system consists of a sequence of send events $(1, i, j, c)$ and receive events $(0, i, j, c)$, where the coupon c was sent by node i with recipient node j, or received by node j, supposedly sent by node i. Note that we must have separate send and receive events, to model adversarial control of message delivery.

Let $\{\hat{c}_i^t\}$ be a set of indicators, $\hat{c}_i^t$ indicating the event that the supervisor algorithm of node P_i supplies signal coupons at time t. This is the only information we need about the behavior of $\mathcal{S}_i$.

We write $\Xi(PK, SK, \mathcal{A}, \{\mathcal{U}_i\}, \{\hat{c}_i^t\})$ for the probability distribution on execution traces given the specified public key, secret key, attacker, update algorithms, and supervisor behaviors.

We show first that, assuming the BCM is secure, the attacker can neither detect nor forge alerts (with non-negligible probability) despite its total control over message traffic. This result holds no matter what update algorithm is used by each node; indeed, it holds even if the update algorithm of each node colludes actively with the adversary.

Clearly, so powerful adversaries can prevent any signal from propagating in the network. We give examples of some simple strategies for spreading an alert quickly through the network given some mild constraints on the attacker's behavior.

5.1   Security

Let us begin with the security properties we want our alert-spreading mechanism to have.

Definition 5. *A set of update algorithms $\{\mathcal{U}_i\}$ is* **secure** *if we have:*

1. **Undetectability**: *Informally, it is hard to distinguish executions where signal coupons are injected from executions where they are not injected. Let $\hat{c}_i^{0,t} = 0$ for all i, t and let $\hat{c}_i^{1,t}$ be arbitrary. Then for any PPT algorithms $\mathcal{A}$ and $\mathcal{D}$,*

$$\Pr \left[b = b' \quad \begin{array}{l} (PK, SK, d, s) \leftarrow \mathcal{G}(1^k); \\ b \xleftarrow{\$} \{0,1\}; \\ \xi \xleftarrow{\$} \Xi\left(PK, SK, \mathcal{A}, \{\mathcal{U}_i\}, \{\hat{c}_i^{b,t}\}\right); \\ b' \leftarrow \mathcal{D}(1^k, PK, d, \{\hat{c}_i^{1,t}\}, \xi) \end{array} \right] - \frac{1}{2} \le negl(k).$$

2. **Unforgeability**: *The adversary cannot cause any node to transmit a signal coupon unless one is supplied by a supervisor. Let $\hat{c}_i^t = 0$ for all i, t. Then for any PPT algorithm $\mathcal{A}$,*

$$\Pr \left[\exists (\cdot, \cdot, \cdot, c) \in \xi \land (c \in S_{SK}) \quad \begin{array}{l} (PK, SK, d, s) \leftarrow \mathcal{G}(1^k); \\ \xi \xleftarrow{\$} \Xi\left(PK, SK, \mathcal{A}, \{\mathcal{U}_i\}, \{\hat{c}_i^t\}\right); \end{array} \right] \le negl(k).$$

Security of the alert-spreading mechanism follows immediately from the security of the underlying blind coupon mechanism. The essential idea behind undetectability is that because neither the adversary nor the update algorithms can distinguish between dummy and signal coupons distributed by the supervisor algorithms, there is no test that can detect their presence or absence. For unforgeability, the inability of the adversary and update algorithms to generate a signal coupon follows immediately from the unforgeability property of the BCM.

Theorem 3. *An alert-spreading mechanism is secure if the underlying blind coupon mechanism is secure.*

Proof. We show first undetectability and then unforgeability.

Undetectability. Let $\{\mathcal{U}_i\}$ be a set of update algorithms, $\{\hat{c}_i^{1,t}\}$ be an arbitrary indicator set, and $\mathcal{A}$ and $\mathcal{D}$ be adversaries against undetectability. We construct a PPT adversary $\mathcal{B}$ against indistinguishability.

Using its input, $\mathcal{B}$ simulates the execution of the protocol in the presence of $\mathcal{A}$. The update algorithms are run as usual. The supervisor processes are simulated as follows: Whenever $\hat{c}_i^{1,t}$ is

zero, the process uses $\mathcal{C}(d, d)$ (taking d from $\mathcal{B}$'s input) to produce a dummy coupon. When $\hat{c}_i^{1,t}$ is non-zero, the process uses $\mathcal{C}(y, y)$ to produce a coupon.

When the execution stops, the execution trace is given to $\mathcal{D}$ which outputs a bit b'. $\mathcal{B}$ then outputs b' and stops.

It is now clear that if $y \in S_{SK}$, then the above process leads to execution traces distributed exactly as if the real system was run with the indicator set $\{\hat{c}_i^{1,t}\}$. On the other hand, if $y \in D_{SK}$, then the above process leads to execution traces distributed exactly as if the real system was run with the indicator set $\{\hat{c}_i^{0,t}\}$.

This means that if $\mathcal{D}$ correctly recognizes which indicator set was used, $\mathcal{B}$ will correctly decide if it was given a signal or dummy coupon.

Unforgeability. Let $\{\mathcal{U}_i\}$ be a set of update algorithms, $\{\hat{c}_i^t\}$ be the all-zero indicator set, and $\mathcal{A}$ be an adversary against unforgeability. We construct a PPT adversary $\mathcal{B}$ against unforgeability for the blind coupon mechanism.

Using its input, $\mathcal{B}$ simulates the execution of the protocol in the presence of $\mathcal{A}$. The update and supervise algorithms are run as usual. Note that since $\hat{c}_i^t$ is zero for all i and t, we do not need a signal coupon to simulate the supervise algorithms.

When the execution stops, all the coupons from the execution trace are accumulated into one coupon using the combining algorithm. This final coupon is then output.

The time cost for the accumulation is linear in the length of the execution trace, hence polynomial time in the input length. From the properties of the combining algorithm we know that if a signal coupon is present in the trace, the output coupon will be a signal coupon except with negligible probability. This means that if $\mathcal{A}$ is successfully forges a signal, $\mathcal{B}$ will successfully forge a signal coupon. □

5.2 Performance

It is not enough that the attacker cannot detect or forge alerts: a mechanism that used no messages at all could ensure that. In addition, we want to make some guarantee that if an alert is injected into the system, it eventually spreads to all non-faulty nodes. In the basic model, the adversary can easily block all communications and prevent a signal from spreading, so we must both specify a particular strategy for the nodes' update algorithms and place restrictions on the attacker's ability to inject and discard messages. We give two simple examples of how the blind coupon mechanism might be used in practice. More sophisticated models can also be used; the important thing is that security is guaranteed as long as the spread of coupons is uncorrelated with their contents.

A Synchronous Flooding Model. Consider a **communication graph** with an edge from each node to each other node that it can communicate to. Suppose that at step t, node P_i's update algorithm (a) discards all invalid incoming coupons; (b) combines any remaining coupons with its previous sent coupons and c_i^t; and (c) sends the result to all of its neighbors in the communication graph. Suppose further that nodes are divided into faulty and non-faulty nodes (by arbitrary choice of the attacker), and that every message sent by a non-faulty node to another non-faulty node is delivered intact by the attacker within at most one time unit. If the communication graph after deletion of faulty nodes is strongly connected, every node receives a signal coupon in at most Δ steps after a signal coupon is injected, where Δ is the diameter of the subgraph of non-faulty nodes.

A Simple Epidemic Model. In this model, the communication graph is complete, and at each step a randomly-chosen node chooses a random node to receive its coupon (which does so immediately). The behavior of a node receiving a message is the same as in the synchronous case. Then the number of interactions from the injection of the first signal coupon until all nodes possess a signal coupon is easily seen to be $O(n \log n)$. Formally:

Theorem 4. *Consider an execution ζ with n nodes of which $b < n$ are Byzantine, and suppose that some sentinel node begins sending a signal at the first step. Let the schedule be determined by choosing pairs of nodes for each step uniformly at random. Then all non-faulty nodes update their state to a signal coupon within expected $O(\frac{n^2 \log n}{n-b})$ steps.*

Proof. First observe that we can assume $b < n - 1$, or else the unique non-faulty node possesses the alert at time 1.

Define a node as "alerted" if its state is a signal coupon, and let k be the number of alerted nodes. If the next step pairs an alerted, non-faulty node with a non-alerted, non-faulty node, which occurs with probability $\frac{k(n-b-k)}{n(n-1)}$, the number of alerted nodes rises to $k+1$. The expected time until this event occurs is at most $\frac{n(n-1)}{k(n-b)} < \frac{n^2}{k(n-b-k)}$. The expected time until all non-faulty nodes are alerted is thus at most

$$
\begin{aligned}
\sum_{k=1}^{n-b-1} \frac{n^2}{k(n-b-k)} &\leq n^2 \left(\sum_{k=1}^{\lceil \frac{n-b-1}{2} \rceil} \frac{1}{k \frac{n-b-1}{2}} + \sum_{k=\lfloor \frac{n-b-1}{2} \rfloor}^{n-b-1} \frac{1}{\frac{n-b-1}{2}(n-b-k)} \right) \\
&\leq 2n^2 \frac{2}{n-b-1} \sum_{k=1}^{\lceil \frac{n-b-1}{2} \rceil} \frac{1}{k} \\
&= \frac{4n^2}{n-b-1} H\left(\left\lceil \frac{n-b-1}{2} \right\rceil \right) \\
&= O\left(\frac{n^2 \log n}{n-b} \right).
\end{aligned}
$$

$\sqcap$

If b is any constant fraction of n, the bound becomes simply $O(n \log n)$.

6   Generic Security of the Subgroup Escape Problem

We now prove that the subgroup escape problem is hard in the generic group model [42]. We first deal with the case from Sect. 4.1 where the representation set is much larger than the group.

Let G be a finite cyclic group and let U be a set such that $|U| \geq |G|$. In the generic group model, elements of G are encoded as unique random elements of U. We define a random injective function $\sigma : G \to U$, which maps group elements to their representations. Algorithms have access to an oracle that on input $x \pm y$ returns $\sigma(\sigma^{-1}(x) \pm \sigma^{-1}(y))$ when both $x, y \in \sigma(G) \subseteq U$, and otherwise the special symbol $\perp$. An algorithm can use the oracle to decide whether $x \in U$ is in $\sigma(G)$ or not by sending the query $x + x$ to the oracle. If $x \notin \sigma(G)$, the reply will be $\perp$.

Theorem 5. *Let D be a subgroup of $G \subseteq U$. Let g be a generator of D. Let $\mathcal{A}$ be a generic algorithm that solves the subgroup escape problem. If $\mathcal{A}$ makes at most q queries to the group oracle, then*

$$\Pr\left[y \in G \setminus D \quad y \leftarrow \sigma^{-1}(\mathcal{A}(1^k, \sigma(g))) \right] \leq \frac{q(|G| - |D|)}{(|U| - 2q - 1)}.$$

Proof. The algorithm can only get information about σ through the group oracle. If the input to the oracle is two elements known to be in $\sigma(D)$, then the adversary learns a new element in $\sigma(D)$. To have any chance of finding an element of $\sigma(G \setminus D)$, the adversary must use the group oracle to test elements that are not known to be in $\sigma(D)$.

Suppose that after i queries, the adversary knows a elements in $\sigma(D)$ and b elements of $U \setminus \sigma(G)$. Since at most two elements are submitted for each query, we know that $a + b \leq 2i + 1$. For any z outside the set of tested elements, the probability that $z \in \sigma(G \setminus D)$ is exactly $(|G| - |D|)/(|U| - b)$ (note that it is independent of a). Therefore, the probability that the adversary discovers an element in $\sigma(G \setminus D)$ with $i + 1$ query is at most $(|G| - |D|)/(|U| - 2i - 1)$. For up to q queries, the probability that at least one of the tested elements are in $\sigma(G \setminus D)$ is at most

$$\sum_{i=1}^{q} \frac{|G| - |D|}{|U| - 2i - 1} \leq q \cdot \frac{|G| - |D|}{|U| - 2q - 1}.$$

$\sqcap$

For a sufficiently large universe U, the escape probability given by Theorem 5 is negligible. The theorem models the construction discussed in Sect. 4.1, but in Sect. 4.2, U is a cyclic group with known group order, and G is a subgroup of U of unknown order. Obviously, we cannot prove an unconditional theorem in the generic model, since it is always possible to factor the group order. When the factorization is known, finding elements of $G \setminus D$ is easy. However, this is essentially the only generic method to solve the subgroup escape problem.

Theorem 6. *Let U be a cyclic group of known composite order $n = \ell_1 \ell_2 \ell_3$, let D be the subgroup of order ℓ_1, and let G be the subgroup of order $\ell_1 \ell_2$. Let $\mathcal{A}$ be a generic algorithm that solves the subgroup escape problem with probability ϵ using time t and q queries to the group oracle. Then there exists an algorithm $\mathcal{B}$ that splits n with probability at least $\epsilon - |G \setminus D|/(|U| - 3q - 2)$ using time $t + O(q^2)$.*

Proof. In order to use $\mathcal{A}$ as a subroutine, we need to simulate both its group oracle and its input. To do that, we need to do arithmetic in the additive group $\mathbb{Z}_n$, but we also need to give the adversary an element of a proper, non-trivial subgroup of $\mathbb{Z}_n$. Since we do not know the factors of n, this is not immediately possible. We shall instead work with the additive group $\mathbb{Z}_n \times \mathbb{Z}_n$. Let γ be an element of $\mathbb{Z}_n$ of order ℓ_1. Define the map $\phi : \mathbb{Z}_n \times \mathbb{Z}_n \to \mathbb{Z}_n$ by $\phi(\xi, \alpha) = \xi + \alpha\gamma$. It is clear that ϕ is a surjective group homomorphism, so $(\mathbb{Z}_n \times \mathbb{Z}_n)/\ker\phi$ is isomorphic to $\mathbb{Z}_n$.

We want to show that if we have two distinct representatives of the same residue class in $(\mathbb{Z}_n \times \mathbb{Z}_n)/\ker\phi$, this will reveal a non-trivial factor of n. Suppose $\phi(\xi, \alpha) = \phi(\xi', \alpha')$. If $\alpha \equiv \alpha' \pmod{\ell_1}$, we have two cases: either $\alpha = \alpha'$, in which case $\xi = \xi'$, or $\alpha = \alpha'$, in which case ℓ_1 divides $\gcd(n, \alpha - \alpha')$ which in turn does not equal n. On the other hand, if $\alpha \not\equiv \alpha' \pmod{\ell_1}$, then $(\alpha - \alpha')\gamma = \xi' - \xi$, hence $\xi' - \xi$ has order ℓ_1 and $\gcd(n, \xi' - \xi) = \ell_2 \ell_3$.

We now describe the factoring algorithm: $\mathcal{B}$ runs $\mathcal{A}$ on input n and a random non-zero number g from $\{0, 1, \ldots, n-1\}$, representing a generator for the subgroup D.

Our algorithm $\mathcal{B}$ must simulate $\mathcal{A}$'s group oracle. It does this with a partial injective mapping $\sigma : \mathbb{Z}_n \times \mathbb{Z}_n \to \{0, 1, \ldots, n-1\}$. Initially, σ is defined only at $(0,0)$ and $(0,1)$, with $\sigma(0,0) = 0$ and $\sigma(0,1) = g$. The mapping is extended as $\mathcal{A}$ queries its group oracle.

If $\mathcal{A}$ queries its oracle with a value x that is not in the image of σ, we choose a random value ξ such that $\sigma(\xi, 0)$ is undefined and extend σ with $\sigma(\xi, 0) = x$.

If $\mathcal{A}$ queries its oracle with $x_0 \pm x_1$, where both values are in the image of σ, but σ is not defined at $\sigma^{-1}(x_0) \pm \sigma^{-1}(x_1)$, we choose a random value x_2 that is not in the image of σ and extend σ with $\sigma(\sigma^{-1}(x_0) \pm \sigma^{-1}(x_1)) = x_2$.

Before we extend σ by $\sigma(\xi, \alpha) = x$, we go through every pair (ξ', α') where σ is defined and compute $\gcd(n, \xi - \xi')$ and $\gcd(n, \alpha - \alpha')$. If either computation produces a non-trivial factor of n, $\mathcal{B}$ stops and outputs a splitting of n.

Eventually, $\mathcal{A}$ stops and outputs x. If x is not in the image of σ, then $\mathcal{B}$ stops and outputs 1. Otherwise, suppose $\sigma^{-1}(x) = (\xi, \alpha)$. Then $\mathcal{B}$ outputs $\gcd(n, \xi)$.

We first need to show that if $\mathcal{A}$ stops, then $\mathcal{B}$ has properly simulated the group oracle. By the initial discussion on the representation of $\mathbb{Z}_n$, it is clear that if $\mathcal{B}$ is about to extend σ non-injectively, it will stop and output a factor. Furthermore, it is clear that when elements are selected at random, they are sampled from the appropriate uniform distributions.

Whenever $\mathcal{A}$ outputs an element that is in the image of σ, and is in $G \setminus D$, say $\sigma(\xi, \alpha) = x$, $\mathcal{B}$ will split n. This is because the order of $\phi(\xi, \alpha)$ must be ℓ_2 or $\ell_1 \ell_2$. Note that $\phi(\xi, \alpha) = \phi(\xi, 0) + \phi(0, \alpha)$. Since the $\phi(0, \alpha)$ has order ℓ_1, $\phi(\xi, 0)$ must have order ℓ_2 or $\ell_1 \ell_2$. Therefore ξ must be divisible by ℓ_3, but not ℓ_2. Hence, $\gcd(n, \xi)$ will be a non-trivial factor of n.

The only case where $\mathcal{A}$ succeeds and $\mathcal{B}$ fails is if $\mathcal{A}$ outputs an element that is not in the image of σ, and this element then happens to be in $G \setminus D$. We need to compute the probability of this event. We note that after q queries to the group oracle, $\mathcal{A}$ has seen at most $3q + 2$ elements in the image of σ. Therefore, the probability that an element not in the image of σ is in $G \setminus D$ is at most $|G \setminus D|/(|U| - 3q - 2)$. This concludes the proof. $\qquad\qquad\square$

7 Related Work

Our motivating example of spreading alerts is related to the problem of anonymous communication. Below, we describe known mechanisms for anonymous communication, and contrast their properties with what can be obtained from the blind coupon mechanism. We then discuss literature on elliptic curves over a ring, which are used in our constructions.

7.1 Anonymous Communication

Two basic tools for anonymous message transmission are DC-nets ("dining-cryptographers" nets) [11, 20] and mix-nets [10]. These tools try to conceal who the message sender and recipient are from an adversary that can monitor all network traffic. While our algorithms likewise aim to hide who the signal's originators are, they are much less vulnerable to disruption by an active adversary that can delay or alter messages, and they can also hide the fact that a signal is being spread through the network.

DC-nets enable one participant to anonymously broadcast a message to others by applying a dining cryptographers protocol. A disadvantage of DC-nets for unstructured systems like peer-to-peer networks is that they require substantial setup and key management, and are vulnerable to jamming. In contrast, the initialization of our alert-spreading application involves distributing only a public key used for verification to non-sentinel nodes and requires only a single secret key shared between the sentinels and the receiver, jamming is prevented by the verification algorithm, and outsiders can participate in the alert-spreading (although they cannot initiate an alert), which further helps disguise the true source. As the signal percolates across the network, all nodes change to an alert state, further confounding the identification of an alert's primary source even if a secret key becomes compromised.

The problem of hiding the communication pattern in the network was first addressed by Chaum [10], who introduced the concept of a **mix**, which shuffles messages and routes them, thereby confusing

traffic analysis. This basic scheme was later extended in [45, 46]. A further refinement is a **mix-net** [1, 21, 22], in which a message is routed through multiple trusted mix nodes, which try to hide correlation between incoming and outgoing messages. Our mechanism is more efficient and produces much stronger security while avoiding the need for trusted nodes; however, we can only send very short messages.

Beimel and Dolev [4] proposed the concept of buses, which hide the message's route amidst dummy traffic. They assume a synchronous system and a passive adversary. In contrast, we assume both an asynchronous system and a very powerful adversary, who in addition to monitoring the network traffic, controls the timing and content of delivered messages.

7.2 Elliptic Curves over a Ring

One of our BCM construction is based on elliptic curves over the ring $\mathbb{Z}_n$, where $n = pq$ is a product of primes. Elliptic curves over $\mathbb{Z}_n$ have been studied for nearly twenty years and are used, *inter alia*, in Lenstra's integer factoring algorithm [29] and the Goldwasser-Kilian primality testing algorithm [18]. Other works [14, 25, 35] exported some factoring-based cryptosystems (RSA [39], Rabin [38]) to the elliptic curve setting in hopes of avoiding some of the standard attacks. The security of our BCM relies on a special feature of the group of points on elliptic curves modulo a composite: It is difficult to find new elements of the group except by using the group operation on previously known elements. This problem has been noted many times in the literature, but was previously considered a nuisance rather than a cryptographic property. In particular, Lenstra [29] chose the curve and the point at the same time, while Demytko [14] used twists and x-coordinate only computations to compute on the curve without y-coordinates. To the best of our knowledge, this problem's potential use in cryptographic constructions was first noted in [16].

Our other BCM construction uses bilinear groups of composite order. Similar groups have been used by Boneh-Goh-Nissim [7] to construct a cryptosystem with interesting homomorphic properties. Their proof of security also relies on the difficulty of the subgroup membership problem, which instills further confidence in our hardness assumptions.

7.3 Epidemic Algorithms

Our alert mechanism belongs to the class of epidemic algorithms (also called gossip protocols) introduced in [13]. In these algorithms, each node randomly chooses neighbors with which to communicate. A drawback of gossip protocols is the number of messages they send, which is in principle unbounded if there is no way for the participants to detect when all information has been fully distributed.

8   Conclusion

We have defined and constructed a blind coupon mechanism, implementing a specialized form of a signed, AND-homomorphic encryption. Our proofs of security are based on the novel subgroup escape problem, which seems hard on certain groups given the current state of knowledge. Our scheme can be instantiated with elliptic curves over $\mathbb{Z}_n$ of reasonable size which makes our constructions practical. We have demonstrated that the BCM has many natural applications. In particular, it can be used to spread an alert undetectably in a variety of epidemic-like settings despite the existence of Byzantine nodes and a powerful, active adversary.

9   Acknowledgments

References

1. M. Abe. Mix-networks on permutation networks. In *Advances in Cryptology - ASIACRYPT '99*, volume 1706 of *Lecture Notes in Computer Science*, pages 258–273. Springer-Verlag, 1999.

2. J. Algesheimer, J. Camenisch, and V. Shoup. Efficient computation modulo a shared secret with applications to the generation of shared safe prime products. In *Advances in Cryptology - Proceedings of CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 417–432. Springer-Verlag, 2002.

3. E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. *Recommendation for key management. Part 1: (revised March 2007)*. National Institute of Standards and Technology, 2007.

4. A. Beimel and S. Dolev. Buses for anonymous message delivery. In *Second International Conference on FUN with Algorithms*, pages 1–13. Carleton Scientific, 2001.

5. I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic Curves in Cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

6. M. Blum, A. D. Santis, S. Micali, and G. Persiano. Non-interactive zero knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991.

7. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Proceedings of Second Theory of Cryptography Conference (TCC 2005)*, pages 325–341, 2005.

8. D. Boneh, K. Rubin, and A. Silverberg. Finding composite order ordinary elliptic curves using the cocks-pinch method. Cryptology ePrint Archive, Report 2009/533, 2009. http://eprint.iacr.org/.

9. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

10. D. Chaum. Untraceable electronic mail, return address and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

11. D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

12. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In L. R. Knudsen, editor, *Proceedings of EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer-Verlag, 2002.

13. A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In F. B. Schneider, editor, *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, pages 1–12, Vancouver, BC, Canada, Aug. 1987. ACM Press.

14. N. Demytko. A new elliptic curve based analogue of RSA. In *Advances in Cryptology - Proceedings of EUROCRYPT 93*, volume 765 of *Lecture Notes in Computer Science*, pages 40–49. Springer-Verlag, 1993.

15. S. D. Galbraith. Elliptic curve Paillier schemes. *Journal of Cryptology*, 15(2):129–138, 2002.

16. K. Gjøsteen. *Subgroup membership problems and public key cryptosystems*. PhD thesis, NTNU, May 2004.

17. K. Gjøsteen. Symmetric subgroup membership problems. In S. Vaudenay, editor, *Proceedings of Public Key Cryptography 2005*, volume 3386 of *LNCS*, pages 104–119. Springer-Verlag, 2005.

18. S. Goldwasser and J. Kilian. Primality testing using elliptic curves. *Journal of the Association for Computing Machinery*, 46:450–472, 1999.

19. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, April 1984.

20. P. Golle and A. Juels. Dining cryptographers revisited. In *Advances in Cryptology - Proceedings of EUROCRYPT 2004*, pages 456–473, 2004.

21. M. Jakobsson. A practical Mix. In *Advances in Cryptology - Proceedings of EUROCRYPT 98*, volume 1403 of *Lecture Notes in Computer Science*, pages 448–461. Springer-Verlag, 1998.

22. M. Jakobsson. Flash mixing. In *Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 83–89. ACM, 1999.

23. R. Johnson, D. Molnar, D. X. Song, and D. Wagner. Homomorphic signature schemes. In *CT-RSA*, pages 244–262, 2002.

24. A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *ANTS*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

25. K. Koyama, U. M. Maurer, T. Okamoto, and S. A. Vanstone. New public-key schemes based on elliptic curves over the ring z_n. In *Advances in Cryptology - Proceedings of CRYPTO 91*, volume 576 of *Lecture Notes in Computer Science*, pages 252–266, 1992.

26. N. Kunihiro and K. Koyama. Equivalence of counting the number of points on elliptic curve over the ring Z_n and factoring n. In Nyberg [34].

27. L. Lamport, R. Shostack, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

28. G.-J. Lay and H. G. Zimmer. Constructing elliptic curves with given group order over large finite fields. In L. M. Adleman and M.-D. A. Huang, editors, *ANTS*, volume 877 of *Lecture Notes in Computer Science*, pages 250–263. Springer-Verlag, 1994.

29. H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649–673, 1987.

30. A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

31. V. S. Miller. The Weil pairing, and its efficient calculation. *J. Cryptology*, 17(4):235–261, 2004.

32. D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. In Nyberg [34], pages 308–318.

33. J. M. G. Nieto, C. Boyd, and E. Dawson. A public key cryptosystem based on the subgroup membership problem. In S. Quing, T. Okamoto, and J. Zhou, editors, *Proceedings of ICICS 2001*, volume 2229 of *Lecture Notes in Computer Science*, pages 352–363. Springer-Verlag, 2001.

34. K. Nyberg, editor. *Advances in Cryptology - EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

35. T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In Nyberg [34], pages 308–318.

36. H. Ong, C.-P. Schnorr, and A. Shamir. An efficient signature scheme based on quadratic equations. In *proceedings of ACM Symposium on Theory of Computing*, ACM, pages 208–216, 1984.

37. P. Paillier. Public-key cryptosystems based on composite degree residue classes. In J. Stern, editor, *Proceedings of EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 1999.

38. M. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Laboratory for Computer Science, Massachusetts Institute of Technology, January 1979.

39. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

40. C. P. Schnorr and J. Pollard. An efficient solution of the congruence $x^2 + ky^2 \equiv m \pmod{n}$. *IEEE Transactions on Information Theory*, 33(5):702–709, 1987.

41. R. Schoof. Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux*, 7:219–254, 1995.

42. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Proceedings of EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer-Verlag, 1997.

43. J. H. Silverman. *The Arithmetic of Elliptic Curves*, volume 105 of *Graduate Texts in Mathematics*. Springer-Verlag, 1986.

44. J. H. Silverman. Computing rational points on rank 1 elliptic curves via L-series and canonical heights. *Mathematics of computation*, 68(226):835–858, April 1999.

45. P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and Onion routing. *IEEE Journal on Selected Areas in Communications: Special Issue on Copyright and Privacy Protection*, 16(4):482–494, 1998.

46. P. F. Syverson, M. G. Reed, and D. M. Goldschlag. Onion routing access configurations. In *DIS-CEX2000:Proceedings of the DARPA information survivability conference and exposition*, pages 34–40. IEEE CS Press, 2000.