Spreading Alerts Quietly and the Subgroup Escape Problem*

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

[1] Yale University, Department of Computer Science,
51 Prospect Street, New Haven, CT 06520, USA,
{aspnes,diamadi,peralta,yampolsk}@cs.yale.edu

[2] Norwegian University of Science and Technology,
Department of Telematics, 7491 Trondheim, Norway,
kristian.gjosteen@item.ntnu.no

Abstract. We introduce a new cryptographic primitive called the **blind coupon mechanism** (BCM). In effect, the BCM is an authenticated bit commitment scheme, which is AND-homomorphic. It has not been known how to construct such commitments before. We show that the 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 Escape Problem, Elliptic Curves Over Composite Moduli, Anonymous Communication.

1 Introduction

MOTIVATION. As more computers become interconnected, chances increase greatly that an attacker may attempt to compromise your 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 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.

---

* An extended abstract of this paper will appear in the proceedings of ASIACRYPT 2005, December 2005.

** Supported in part by NSF grants CCR-0098078, CNS-0305258, and CNS-0435201.

*** Supported by NSF grants CCR-0098078, ANI-0207399, CNS-0305258, and CNS-0435201.

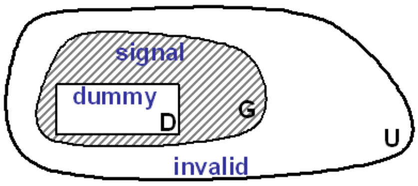

Fig. 1. Abstract group structure used in our BCM construction.

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 processes 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, yet quite different, from the notion of 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 if 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.

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 new signal coupon unless he is given another signal coupon as input (**unforgeability**); and (c) cannot distinguish randomly chosen coupons from coupons produced by the combining algorithm (**blinding**).

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 subgroup of G. The elements of D will represent dummy coupons and the elements of $G \setminus D$ will be signal coupons (see also Figure 1). The combining operation will simply be a group operation. To make verification possible, there will need to be an easy way to distinguish elements of G (valid coupons) from elements of $U \setminus G$ (invalid coupons).

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 an element $y \in G$, decide whether $y \in D$ or $y \in G \setminus D$.

– **Subgroup Escape Problem**: Given a generator for D (but not G), find an element of $G \setminus D$.

The subgroup membership problem has appeared in many different forms in the literature [CS02, GM84, NS98, OU98, Pai99, Gjø05, NBD01]. The subgroup escape problem has not been studied before. To provide more confidence in its validity, we later analyze it 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. 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? The BCM can potentially be useful in various applications. If signal coupons are used to encode a "1" and dummy coupons a "0", then a BCM can be viewed as an OR-homomorphic bit commitment scheme. The BCM 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. In addition, the BCM's verification function ensures the commitment is authenticated. By switching signal coupons to encode a "0" and dummy coupons to encode a "1", we get an AND-homomorphic bit commitment. As far as we know, it has not been known how to construct such commitments before. The BCM thus provides a missing link in protocol design. Using BCM together with techniques of Brassard *et al.* [BCC88], we can obtain short non-interactive proofs of circuit satisfiability, whose length is linear in the number of AND gates in the circuit. Other potential uses include i-voting (voting over the Internet) [CRS04].

SPREADING ALERTS WITH THE BCM. Returning to our original motivation, we demonstrate how a BCM can be used to propagate alerts quickly and quietly throughout the network. 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 [LSP82]: 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 with a discussion of related work in Section 2. 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. Then 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. Conclusions and open problems appear in Section 7.

2 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.

2.1 Anonymous Communication

Two basic tools for anonymous message transmission are DC-nets ("dining-cryptographers" nets) [Cha88, GJ04] and mix-nets [Cha81]. 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 [Cha81], who introduced the concept of a **mix**, which shuffles messages and routes them, thereby confusing traffic analysis. This basic scheme was later extended in [SRG00, SGR98]. A further refinement is a **mix-net** [Abe99, Jak99, Jak98], 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 small messages.

Beimel and Dolev's [BD01] 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 very powerful adversary, who in addition to monitoring the network traffic controls the timing and content of delivered messages.

2.2 Elliptic Curves over a Ring

One of our BCM constructions 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 [HWL87] and the Goldwasser-Kilian primality testing algorithm [GK99]. Other works [Dem93, KMOV92, OU98] exported some factoring-based cryptosystems (RSA [RSA78], Rabin [Rab79]) 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 [HWL87] chose the curve and the point at the same time, while Demytko [Dem93] 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 [Gjø04].

2.3 Epidemic Algorithms

Our alert mechanism belongs to the class of epidemic algorithms (also called gossip protocols) introduced in [DGH$^+$87]. In these algorithms, each process chooses to partner processes with which to communicate randomly. The 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.

3 Blind Coupon Mechanism

The critical component of our algorithms that allows information to propagate undetectably among the processes 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 the 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:*

x	y	$\mathcal{C}(x,y)$
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.

- *$\mathcal{G}(1^k)$, the probabilistic **key generation algorithm**, outputs a pair of public and secret keys (PK, SK) and two strings (d, s). The public key defines a universe set U_{PK} and a set of **valid coupons** G_{PK}. The secret key implicitly defines an associated set of **dummy coupons** D_{SK} and a set of **signal coupons** S_{SK}.[3] It is the case that $d \in D_{SK}$ 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.*
- *$z \leftarrow \mathcal{C}_{PK}(x, y)$, the probabilistic **combining algorithm**, takes as input two valid coupons $x, y \in G_{PK}$ and produces a new coupon z. The output z is a signal coupon (with overwhelming probability) whenever one or more of the inputs is a signal coupon, otherwise it is a dummy coupon (see Figure 2).*
- *$\mathcal{D}_{SK}(y)$, the deterministic **decoding algorithm**, takes as input a valid coupon $y \in G_{PK}$. It returns 0 if y is a dummy coupon and 1 if y is a signal coupon.*

The BCM may be established either by an external trusted party or jointly by the application participants, running the distributed key generation protocol (*e.g.*, one could use a variant of [ACS02]). In this paper, we assume a trusted dealer (the network administrator) who runs the key generation algorithm and distributes signal coupons to the supervisor algorithms of 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 $\mathcal{V}_{PK}$ prevents the adversary from flooding the system with invalid coupons and making it impossible for the signal to spread.

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*

[3] Note that membership in S_{SK} and D_{SK} should not be efficiently decidable when given only PK (unlike membership in G_{PK}). However, we require that membership is always efficiently decidable when given SK.

PPT algorithm $\mathcal{A}$,

$$\left| \Pr\left[b = b' \;\middle|\; \begin{array}{c} (PK, SK, d, s) \leftarrow \mathcal{G}(1^k); \\ x_0 \xleftarrow{\$} D_{SK}; x_1 \xleftarrow{\$} S_{SK}; \\ b \xleftarrow{\$} \{0,1\}; b' \leftarrow \mathcal{A}(1^k, PK, d, x_b) \end{array} \right] - \frac{1}{2} \right| \leq negl(k)$$

2. ***Unforgeability**: The adversary is unlikely to fabricate a signal coupon without the use of another signal coupon as input[4]. Formally, for any PPT algorithm $\mathcal{A}$,*

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

3. ***Blinding**: The combination $\mathcal{C}_{PK}(x,y)$ of two valid coupons x,y looks like a random valid coupon. Formally, fix some pair of keys (PK, SK) outputted by $\mathcal{G}(1^k)$. Let U_D be a uniform distribution on D_{SK} and let U_S be a uniform distribution on S_{SK}. Then, for all valid coupons $x, y \in G_{PK}$,*

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

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

To build the reader's intuition, we describe a straw-man construction of a BCM. Suppose we are given any semantically secure encryption scheme $\mathcal{E}(\cdot)$ and a set-homomorphic signature scheme $\mathrm{SIG}(\cdot)$ by Johnson *et al.* [JMSW02]. 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 random-length vector of encrypted zeroes; *e.g.*, $x = (\mathcal{E}(0), \ldots, \mathcal{E}(0))$. The 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: $\mathcal{C}_{PK}\left((x, \mathrm{SIG}(x)), (y, \mathrm{SIG}(y))\right) = \left(x \cup y, \mathrm{SIG}(x \cup y)\right)$. 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), where U is a finite set, and G is a subset of U. G also has a group structure: it is a cyclic group generated by s.

[4] 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.

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; moreover, $|G|/|U| \leq negl(k)$ and $|D|/|G| \leq negl(k)$.

Let $\mathcal{G}'$ be a PPT algorithm that on input of 1^k samples from Γ_k according to some distribution. We consider Γ_k to be a probability space with this distribution.

We assume there exists 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.

- 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** $\mathcal{D}_{SK}$ checks if $|D| \cdot y = 0$.

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

Definition 3. *The* ***subgroup membership problem*** *for* Γ *asks: given a tuple* (U,G,D,d,s) *from* Γ *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 $\mathcal{A}$,

$$\left| \Pr\left[b' = b \;\middle|\; \begin{array}{c} (U,G,D,d,s) \xleftarrow{\$} \Gamma_k; \\ y_0 \xleftarrow{\$} D; y_1 \xleftarrow{\$} G \setminus D; \\ b \xleftarrow{\$} \{0,1\}; b' \leftarrow \mathcal{A}(U,G,D,d,s,y_b) \end{array} \right] - \frac{1}{2} \right| \leq negl(k).^5$$

Various subgroup membership problems have been extensively studied in the literature, and examples include the Decision Diffie-Hellman problem [CS02], the quadratic residue problem [GM84], among others [NS98, OU98, Pai99]. Our constructions however are more related to the problems described in [Gjø05, NBD01].

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

[5] Henceforth, we assume that groups we operate on have some concise description, which can be passed as an argument to our algorithms. We also assume that group elements can be uniquely encoded as bit strings.

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

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

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.

Theorem 1. *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 prove the blinding property first, and start with the ideal case: For input $x, y \in G$, sample r_0 uniformly from $\{0,1,\ldots,|D|-1\}$, and r_1 and r_2 uniformly from $\{0,1,\ldots,|G/D|-1\}$, and output $r_0 g + r_1 x + r_2 y$.

If $x, y \in D$, the product is uniformly distributed in D, since $r_0 g$ is.

If $x \notin D$, then the residue class $r_1 x + D$ is uniformly distributed in G/D. Since $r_0 g$ is uniformly distributed in D, the product is uniformly distributed in G. The uniform distribution on G is $|D|/|G|$-close to the uniform distribution on $G \setminus D$. The same argument holds for $r_2 y$.

Finally we note that we do not need to know $|D|$ or $|G/D|$. Since we know that $|D|$ and $|G/D|$ are less than 2^k, sampling r_0, r_1, r_2 uniformly from the set $\{0,\ldots,2^{2k}-1\}$ will produce an output distribution that is 2^{-k}-close to ideal, which proves the bound for blinding

Next, we prove the indistinguishability property, so let $\mathcal{A}$ be an adversary against indistinguishability. We have a subgroup membership problem instance (U,G,D,d,s) and $y \in G$. We construct the public key $PK = (U,G,d,k)$, and give $\mathcal{A}$ as input PK, d and y.

If $\mathcal{A}$ answers 1, we conclude that $y \in G \setminus D$, otherwise $y \in D$. Whenever $\mathcal{A}$ is correct, we will be correct, so $\mathcal{A}$ must have negligible advantage.

Finally, we deal with forging. Let $\mathcal{A}$ be an adversary against unforgeability. We have a subgroup escape problem instance U, G and D, and a generator d for D. Again we construct the public key $PK = (U,G,d,k)$, and give $\mathcal{A}$ as input PK and d.

Our output is simply $\mathcal{A}$'s output. Whenever $\mathcal{A}$ succeeds, we will succeed, so $\mathcal{A}$ must have negligible success probability. □

4 Constructing the BCM

We now give two instantiations of the abstract group structure (U,G,D) described in the previous section. First, we review some basic facts about elliptic curves over composite moduli in Section 4.1. Then, in Section 4.2, we describe our BCM construction that utilizes these curves. In Section 4.3, we describe an alternative BCM construction on elliptic curves equipped with bilinear pairings. These constructions can be used to undetectably transmit a one-shot signal throughout the network. In Section 4.4, we describe how the BCM's bandwidth can be further expanded.

4.1 Preliminaries

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 $\bar{U}$ of triples $(x,y,z) \in \mathbb{Z}_n^3$ satisfying $\gcd(x,y,z,n) = 1$. Let $\sim$ be the equivalence relation on $\bar{U}$ defined by $(x,y,z) \sim (x',y',z')$ iff there exists $\lambda \in \mathbb{Z}_n^*$ such that $(x,y,z) = (\lambda x', \lambda y', \lambda z')$. Let U be the set of equivalence classes in $\bar{U}$. 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 $\gcd(4a^2 - 27b^3, n) = 1$. The set of points on $E/\mathbb{Z}_n$ is the set of equivalence classes $(x:y:z) \in U$ 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 $E_p : Y^2 Z = X^3 + a_p XZ^2 + b_p Z^3$ and $E_q : Y^2 Z = X^3 + a_q XZ^2 + b_q Z^3$ 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.

It can also be shown that the Chinese remainder theorem gives a set isomorphism

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

inducing 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 [Gal02] for more details).

4.2 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 [BSS99, LZ94].

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 $SK = (p, q, l_1, l_2, l_3)$.[6]

[6] To describe 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 $y^2z \equiv x^3 + axz^2 + bz^3 \pmod{n}$.

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 [Gjø05].

Counting the number of points on an elliptic curve defined over a composite number is equivalent to factoring the number [HWL87, KK98]. Therefore, the group order $E_p(\mathbb{F}_p)$ is hidden.

When the group order is hidden, it cannot be factored. It therefore seems reasonable 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 [SP87] gave an algorithm to find solutions of a quadratic equation modulo a composite (which broke the Ong-Schnorr-Shamir signature system [OSS84]). 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 [Sil99]. 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. We can find the fourth intersection point P_4 by deriving a fourth-degree polynomial in X for which we know three zeros.

To show that we could easily derive this point using the group operation, we consider the situation over the finite fields, where E and C have at most six points of intersection. Both intersect $(0:1:0)$, and since the line $Z = 0$ is a tangent to both curves in $(0:1:0)$, their intersection number in $(0:1:0)$ is greater than 1. This means that E and C intersect in exactly five points, P_1, P_2, P_3, P_4 and $(0:1:0)$.

The divisor of C is $(P_1)+(P_2)+(P_3)+(P_4)+2((0:1:0))$. Let $C' : Z^2 = 0$ with divisor $6((0:1:0))$. Since the divisor of the function $f(X,Y,Z) = (YZ - \alpha X^2 - \beta XZ - \gamma Z^2)/(Z^2)$ satisfies $\mathrm{div}(f) = \mathrm{div}(C) - \mathrm{div}(C') = 0$, we see that $(P_1)+(P_2)+(P_3)+(P_4) - 4((0:1:0)) = 0$, which means that

$$P_1 + P_2 + P_3 + P_4 = (0:1:0)$$

The fourth point is therefore 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 in the previous section, yield a hard subgroup escape problem.

4.3 BCM on Groups With Bilinear Pairings

Let p, ℓ_1, ℓ_2, and ℓ_3 be primes such that $p+1 = 6\ell_1\ell_2\ell_3$, and $p = 2 \pmod 3$. 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 [BF01].

Let $U = E(\mathbb{F}_p)$, and let G and D be the subgroups of $E(\mathbb{F}_p)$ of order $\ell_1\ell_2$ 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 $SK = (l_1, l_2, l_3)$. 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 [Gjø05] 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.

4.4 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 $x = x_1 \dots x_m$ is represented by a vector of coupons $\langle c_1, \dots, c_{2m} \rangle$, where each c_i is drawn from a different scheme. Each processor applies his 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. We can enable the receiver to detect when it has received a complete message by representing each bit x_i by two coupons: c_{2i-1} (for $x_i = 0$) and c_{2i} (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.

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 $x = x_1 \dots x_m$ 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 $r_1, \dots, r_m \in \{0, 1, \dots, 2^{2k} - 1\}$.

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 4.2, n_0 is a product of two moderately large primes, while the other primes can be around 2^{80}. For the construction from Section 4.3, n_0 is prime, but every prime must be fairly large to counter elliptic curve factorization.

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 begin with a definition of the problem in Sections 5.1, and then present results on the security and performance of the mechanism in Sections 5.2 and 5.3.

To summarize these results briefly, 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 nodes carry out these operations under the control of a PPT **attacker** $\mathcal{A}$ 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.

We show first that, assuming the BCM is secure, the attacker can neither detect nor forge alerts 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 half of each node colludes actively with the adversary. We then give examples of some simple strategies for spreading an alert quickly through the network with some mild constraints on the attacker's behavior.

5.1 Our Model

We now describe the model for our algorithms.

5.1.1 Basic Setting We adopt a very general message-passing communications model, permitting an active adversary both control over the timing of delivery of messages between nodes and the ability to read, replace, and redirect messages at will. At the same time, we structure our model of a node to enforce the requirement that the node's visible behavior (*e.g.*, its choices of what other nodes to communicate with) is not affected by the type of coupons it is transmitting.

5.1.2 Processes We assume that we have a collection of n nodes $P_1, P_2, \ldots, P_n$. Processes have "split brains": for each node P_i an **update algorithm** $\mathcal{U}_i$ handles communication with other nodes, while a **supervisor algorithm** $\mathcal{S}_i$ chooses when or if to send a signal coupon. This split enforces the requirement that the communication pattern does not depend on which type of coupon a node is sending.

We do not examine the behavior of the supervisor algorithm closely; instead, we assume only that it supplies a sequence of coupons $c_i^1, c_i^2, \ldots$ to the update algorithm $\mathcal{U}_i$. The supervisor algorithm $\mathcal{S}_i$ of regular nodes will intermittently supply a sequence of dummy coupons. Meanwhile, $\mathcal{S}_i$ of sentinel nodes will supply dummy coupons until it detects the intruder's presence, at which point it will switch to dispensing signal coupons. We assume that the sequence does not depend on the execution of the rest of the protocol. For convenience, we write $\hat{c}_i^t$ for the indicator variable that c_i^t is a signal coupon; that is, we write $\hat{c}_i^t = 0$ if at step t of execution the coupon supplied by the supervisor algorithm of node P_i is a dummy coupon and $\hat{c}_i^t = 1$ if it is signal.

The inputs to update algorithm $\mathcal{U}_i$ at step t consist of (a) the sequence of sets of messages received at steps 1 through t; (b) the sequence of sets of messages sent at steps 1 through $t-1$; and (c) the coupon c_i^t supplied by $\mathcal{S}_i$ at time t. The output of $\mathcal{U}_i$ is a set of messages to be sent at step t. Each message is of the form (s, r, m, c) where s is the identity of the sender, r is the intended recipient, m is an arbitrary string, and c is a coupon. To simplify the model, we do not keep track of a separate process state, because any such state can easily be recomputed from the message history.

The update algorithms have access to the public key PK of the blind coupon mechanism. We assume that they can apply the verification algorithm $\mathcal{V}_{PK}$ and the combining algorithm $\mathcal{C}_{PK}$ in computing outgoing messages. To spread alerts, a typical update algorithm will discard any coupons from incoming messages or the supervisor algorithm that are rejected by $\mathcal{V}_{PK}$, and forward to a carefully-chosen set of recipients coupons obtained by combining all unrejected coupons so far in some order using $\mathcal{C}_{PK}$. It may also use additional information in messages to manage spreading of alerts, and this additional information may also depend on the values of the coupons it has seen.

5.1.3 Attacker The PPT attacker algorithm $\mathcal{A}$ controls the timing and content of delivered messages. The input to the attacker is a partial execution, where the t-th step of an execution is described by a tuple (i_t, R_t, S_t) where i_t is a node identity, R_t is the set of messages received by P_{i_t} at that step, and S_t is the set of messages sent by P_{i_t} at that step. The output of $\mathcal{A}$ is a choice of which node $P_{i_{t+1}}$ executes the next step and what set of messages R_{t+1} it receives. The attacker also has access to the public key PK and can use the verification and combining algorithms $\mathcal{V}_{PK}$ and $\mathcal{C}_{PK}$ as subroutines.

An execution is constructed by an interactive protocol which alternates between the attacker choosing a node $P_{i_{t+1}}$ and a set of received messages R_{t+1} and the node's update algorithm $\mathcal{U}_i$ computing a set of messages S_{t+1} to send. Given particular public and secret keys, PK and SK, adversary $\mathcal{A}$, update algorithms $\mathcal{U}_i$, and supervisor inputs $\hat{c}_i^t$ for steps $t = 1, \ldots, T$, there exists a corresponding probability distribution $\Xi(PK, SK, \mathcal{A}, \{\mathcal{U}_i\}, \{\hat{c}_i^t\})$ on executions.

Note that traditional classes of process faults are easily simulated by an attacker defined in this way: a Byzantine node, for example, can be simulated by replacing all of its outgoing messages in transit. The attacker also has full power to violate any assumptions about synchrony, timely delivery, or reliable message transmission that the algorithm makes. We will show that such violations do not affect the security guarantees derived from the blind coupon mechanism; however, any performance guarantees on alert-spreading will require imposing restrictions on the attacker's behavior.

5.1.4 Problem The problem is simple: at an opportune time, the sentinel nodes wish to propagate an alert (signal coupons) to all other nodes. We want to prevent the attacker (except with negligible probability) from (a) identifying the presence or source of signal coupons; (b) causing the nodes to spread signal coupons even though no supervisor algorithm supplied one; (c) preventing the spread of signal coupons to potential recipients.

5.2 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, for any adversary algorithm* $\mathcal{A}$*, and any* $T = poly(k)$*, we have:*

1. ***Undetectability****: Given two distributions on executions, one in which no signal coupons are injected by supervisors and one in which some are, the adversary cannot distinguish between them with probability greater than* $1/2$*. Formally, let* $\hat{c}_i^{0,t} = 0$ *for all i, t and let* $\hat{c}_i^{1,t}$ *be arbitrary. Then for any PPT algorithm* $\mathcal{D}$*,*

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

2. ***Unforgeability****: The adversary cannot cause any process to transmit a signal coupon unless one is supplied by a supervisor. Formally, if* $\hat{c}_i^t = 0$ *for all i, t, then there is no PPT algorithm* $\mathcal{A}$ *such that*

$$\Pr\left[\exists (s,r,m,c) \in \xi \wedge (c \in S_{SK}) \;\middle|\; \begin{array}{c} (PK,SK,d,s) \leftarrow \mathcal{G}(1^k); \\ \xi \xleftarrow{\$} \Xi\left(PK,SK,\mathcal{A},\{\mathcal{U}_i\},\{\hat{c}_i^t\}\right); \end{array} \right] \leq negl(k).$$

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

Proof (sketch). We show first undetectability and then unforgeability.

Undetectability. Suppose that the alert-spreading mechanism does not satisfy undetectability, *i.e.* that there exists a set of update algorithms $\{\mathcal{U}_i\}$, an adversary $\mathcal{A}$, and pattern $\{\hat{c}_i^{1,t}\}$ of signal coupons that can be distinguished from only dummy coupons by some PPT algorithm $\mathcal{D}$ with non-negligible probability.

Let us use this fact to construct a PPT algorithm $\mathcal{B}$ that violates indistinguishability. Let y be the coupon input to $\mathcal{B}$. Then $\mathcal{B}$ will simulate an execution ξ of the alert-spreading protocol by simulating the adversary $\mathcal{A}$ and the appropriate update algorithm $\mathcal{U}_i$ at each step. The only components of the protocol that $\mathcal{B}$ cannot simulate directly are the supervisor algorithms $\mathcal{S}_i$, because $\mathcal{B}$ does not have access to signal coupons provided to the supervisor algorithms of sentinel nodes. But here $\mathcal{B}$ lets $c_i^t = \mathcal{C}(d,d)$ when $\hat{c}_i^{1,t} = 0$ and lets $c_i^t = \mathcal{C}(y,y)$ when $\hat{c}_i^{1,t} = 1$. By the blinding property of the BCM, if $y \in D_{SK}$, then all coupons c_i^t will be statistically indistinguishable from uniformly random dummy coupons, giving a distribution on executions that is itself statistically indistinguishable from $\Xi\left(PK, SK, \mathcal{A}, \{\mathcal{U}_i\}, \{\hat{c}_i^{0,t}\}\right)$. If instead $y \in S_{SK}$, then c_i^t will be such that the resulting distribution on executions will be statistically indistinguishable from $\Xi\left(PK, SK, \mathcal{A}, \{\mathcal{U}_i\}, \{\hat{c}_i^{1,t}\}\right)$. It follows from the indistinguishability property of the BCM that no PPT algorithm $\mathcal{D}$ can distinguish between these two distributions with probability greater than $1/2 + negl(k)$.

Unforgeability. The proof of unforgeability is similar. Suppose that there is some adversary and a set of update functions that between them can, with non-negligible probability, generate a signal coupon given only dummy coupons from the supervisor algorithms. Then a PPT algorithm $\mathcal{B}$ that simulates an execution of this system and returns a coupon obtained by combining all valid coupons sent during the execution forges a signal coupon with non-negligible probability, contradicting the unforgeability property of the BCM.

□

5.3 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. To do so requires both specifying a particular strategy for the nodes' update algorithms and placing restrictions on the attacker's ability to 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 3. *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

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

□

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 prove that the subgroup escape problem is hard in the generic group model [Sho97] when the representation set is much larger than the group.

Let G be a finite cyclic group and let $U \subseteq \{0,1\}^*$ be a set such that $|U| \geq |G|$. In the generic group model, elements of G are encoded as unique random strings. We define a random injective function $\sigma : G \to U$, which maps group elements to their string representations. Algorithms have access to an oracle that on input of $x \pm y$ 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 4. *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*

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

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 $U \setminus \sigma(G)$ $(a + b \leq i)$. 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 $(|G| - |D|)/(|U| - i)$. For up to q queries, the probability that at least one of the tested elements are in $\sigma(G \setminus D)$ is at most

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

For a sufficiently large universe U, this probability is negligible. □

7 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.

8 Acknowledgments

We are grateful to Yevgeniy Dodis for his helpful comments regarding this work.

References

[Abe99] Masayuki Abe. Mix-networks on permutation networks. In *Advances in Cryptology - Proceedings of ASIACRYPT 99*, volume 1706 of *Lecture Notes in Computer Science*, pages 258–273. Springer-Verlag, 1999.

[ACS02] Joy Algesheimer, Jan Camenisch, and Victor 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.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BD01] Amos Beimel and Shlomi Dolev. Buses for anonymous message delivery. In *Second International Conference on FUN with Algorithms*, pages 1–13. Carleton Scientific, 2001.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *Lecture Notes in Computer Science*, 2139:213–229, 2001.

[BSS99] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Elliptic Curves in Cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

[Cha81] David Chaum. Untraceable electronic mail, return address and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[CRS04] David Chaum, Peter Y.A. Ryan, and Steve A. Schneider. A practical, voter-verifiable election scheme. Technical Report CS-TR-880, School of Computing Science, University of Newcastle, December 2004.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Proceedings of EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer-Verlag, 2002.

[Dem93] 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.

[DGH+87] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In Fred B. Schneider, editor, *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, pages 1–12, Vancouver, BC, Canada, August 1987. ACM Press.

[Gal02] Steven D. Galbraith. Elliptic curve Paillier schemes. *Journal of Cryptology*, 15(2):129–138, 2002.

[GJ04] Philippe Golle and Ari Juels. Dining cryptographers revisited. In *Advances in Cryptology - Proceedings of EUROCRYPT 2004*, pages 456–473, 2004.

[Gjø04] Kristian Gjøsteen. *Subgroup membership problems and public key cryptosystems*. PhD thesis, NTNU, May 2004.

[Gjø05] Kristian Gjøsteen. Symmetric subgroup membership problems. In Serge Vaudenay, editor, *Proceedings of Public Key Cryptography 2005*, volume 3386 of *LNCS*, pages 104–119. Springer-Verlag, 2005.

[GK99] Shafi Goldwasser and Joe Kilian. Primality testing using elliptic curves. *Journal of the Association for Computing Machinery*, 46:450–472, 1999.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, April 1984.

[HWL87] Jr. Hendrik W. Lenstra. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649–673, 1987.

[Jak98] Markus 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.

[Jak99] Markus Jakobsson. Flash mixing. In *Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing*, pages 83–89. ACM, 1999.

[JMSW02] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In *CT-RSA*, pages 244–262, 2002.

[KK98] Noboru Kunihiro and Kenji Koyama. Equivalence of counting the number of points on elliptic curve over the ring Z_n and factoring n. In Nyberg [Nyb98].

[KMOV92] Kenji Koyama, Ueli M. Maurer, Tatsuaki Okamoto, and Scott 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.

[LSP82] Leslie Lamport, Robert Shostack, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Proggramming Languages and Systems*, 4(3):382–401, 1982.

[LZ94] Georg-Johann Lay and Horst G. Zimmer. Constructing elliptic curves with given group order over large finite fields. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *ANTS*, volume 877 of *Lecture Notes in Computer Science*, pages 250–263. Springer-Verlag, 1994.

[NBD01] Juan Manuel González Nieto, Colin Boyd, and Ed 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.

[NS98] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In Nyberg [Nyb98], pages 308–318.

[Nyb98] Kaisa Nyberg, editor. *Advances in Cryptology - EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[OSS84] H. Ong, Claus-Peter Schnorr, and Adi Shamir. An efficient signature scheme based on quadratic equations. In *proceedings of ACM Symposium on Theory of Computing*, ACM, pages 208–216, 1984.

[OU98] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In Nyberg [Nyb98], pages 308–318.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residue classes. In Jacques Stern, editor, *Proceedings of EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 1999.

[Rab79] Michael 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.

[RSA78] Ronald Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[SGR98] Paul F. Syverson, David M. Goldschlag, and Michael 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.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Proceedings of EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer-Verlag, 1997.

[Sil99] Joseph 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.

[SP87] Claus P. Schnorr and John M. 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.

[SRG00] Paul F. Syverson, Michael G. Reed, and David M. Goldschlag. Onion routing access configurations. In *DISCEX2000:Proceedings of the DARPA information survivability conference and exposition*, pages 34–40. IEEE CS Press, 2000.