Covert Channels in Privacy-Preserving Identification Systems

Daniel V. Bailey[1], Dan Boneh[2] and Eu-Jin Goh[3], and Ari Juels[4]

[1] RSA Laboratories
dbailey@rsa.com
[2] Stanford University
dabo@cs.stanford.edu
[3] Stanford University
eujin@cs.stanford.edu
[4] RSA Laboratories
ajuels@rsa.com

Abstract. We examine covert channels in privacy-enhanced mobile identification devices where the devices uniquely identify themselves to an authorized verifier. Such devices (e.g. RFID tags) are increasingly commonplace in hospitals and many other environments. For privacy, the device outputs used for identification should "appear random" to any entity other than the verifier, and should not allow physical tracking of device bearers. Worryingly, there already exist privacy breaches for some devices [28] that allow adversaries to physically track users. Ideally, such devices should allow anyone to publicly determine that the device outputs are covert-channel free (CCF); we say that such devices are CCF-checkable.

Our main result shows that there is a fundamental tension between identifier privacy and CCF-checkability; we show that the two properties cannot co-exist in a single system. We also develop a weaker privacy model where a continuous observer can correlate appearances of a given tag, but a sporadic observer cannot. We also construct a privacy-preserving tag identification scheme that is CCF-checkable and prove it secure under the weaker privacy model using a new complexity assumption. The main challenge addressed in our construction is the enforcement of *public verifiability*, which allows a user to verify covert-channel-freeness in her device without managing secret keys external to the device.

1 Introduction

We examine covert channels in the context of *identifier privacy* in privacy-enhanced mobile identification devices. In these settings, a device uniquely identifies itself to an authorized verifier by a sequence of changing output values. The privacy property of the system is that the devices' outputs used for identification should "appear random" to any entity other than the verifier, and thus does not permit physical tracking of device bearers. We use the generic term "tag" in this paper to denote all mobile identification devices.

We note that previous work has focused primarily on covert channels that leak the keys associated with systems concerned with data integrity (digital signatures) or data confidentiality (encryption schemes), whereas we consider identification schemes in this paper.

RFID tags are perhaps the most interesting and pervasive example of identification devices for which such privacy is important. For example, hospitals are deploying RFID tags to identify newborn infants and patients. As demonstrated recently by Saponas et al. [28], privacy breaches for such tags are not merely theoretical—Saponas et al. showed that users of the "Nike+Ipod" Sport Kit can be tracked up to a distance of 60 feet by monitoring the unique identifier that the mobile device (attached to a user's

shoe) emits. More generally, there are several scenarios in which covert channels can be problematic in tags:

1. *Covert Sensors / Transaction Monitors:* A verifier might use a covert channel to extract side information from a tag. For example, the verifier might obtain information from a hidden sensor, or unauthorized information about transactions performed by the tag.
2. *Covert Identification Channels:* A manufacturer could implant a covert channel to create an independently and secretly configurable tracking system. Such a system can operate even without the collusion of authorized verifiers [28].
3. *Covert Authentication:* Paradoxically, it is sometimes desirable for mobile devices like RFID tags to be *cloneable*; that is, not to authenticate themselves. Researchers have argued in favor of this property in human-implantable RFID tags [10]. If an attacker can easily clone such RFID tags, she will have little incentive to steal them; that is, physically extract the tags, thus harming their owners. The presence (or even the mere possibility) of a covert authentication channel can undermine this important assurance of safety.

As these scenarios show, covert channels are particularly problematic in personal wireless devices that users carry over a long time, and also in devices capable of harvesting and leaking sensitive ambient information. Such devices merit particularly strong privacy protection for users. Such devices also are less cost-sensitive and are endowed with more circuitry than cheaper transient devices such as barcode-type RFID [30]; these devices typically have greater computational power, which we exploit for the schemes detailed in this paper.

It is well known that only a *deterministic* protocol can be certifiably free of covert channels. In this paper, we consider privacy preserving tag-identification protocols that are verifiably deterministic. More precisely, we aim to achieve the challenging property of *public verifiability. Our constructions allow any entity to verify covert-freeness in a tag* without having to refer to or manage any secret keying material. *We believe that public verifiability is an essential feature in the useability of our scheme, because key management is a pandemic problem in the management of personal devices, particularly for RFID [15, 23]. Public verifiability of covert-freeness alleviates the need for a user to manage secret keying material external to her tag and even allows third-parties to check covert-freeness in tags. We refer to privacy-preserving identification protocols with the properties we describe here as* covert-channel-free-*checkable, or, for brevity, as* CCF-checkable. Note that we do not consider covert channels at lower layers that are implemented through timing or power usage variations; such channels have been studied in the literature [20] and have well-understood (but perhaps inefficient) solutions.

Our Results.

1. Our main result shows that there is a fundamental tension between identifier privacy and CCF-checkability. Having set forth formal definitions for the two properties, we show that in the strictest sense, they cannot co-exist in a single system.
2. We next consider a natural loosening of the privacy property that will allow a tag to be CCF-checkable. In particular, we construct systems in which a continuous observer can correlate appearances of a given tag, but a sporadic observer cannot. We show that in such systems where privacy is sacrificed in a tightly localized way, it is possible to achieve CCF-checkability.

Note that if we assume that verifiers possessed tag-specific secret keys, it is straightforward to construct covert-checkable identification protocols that meet our privacy requirements. For example, if a device generated outputs using a PRNG, a device owner who holds the corresponding seed can check for correct behavior. Thus a particular challenge in the architecture of our schemes is the enforcement of public verifiability. Again, this property eliminates the need for external key-management by device

owners, allows CCF-checking by third parties, and in general decouples identification from verification of covert-freeness.

Our tag identification scheme strikes a balance between privacy and covert-freeness. We also develop a security model that captures this loosening of privacy for tags, and we prove our construction secure in this model using a new complexity assumption.

Our protocol is somewhat computationally intensive for small devices, as it requires multiplication over bilinear groups of composite order. (Our scheme also uses pairings in a bilinear group, but these are not computed on the tags.) Such operations are well beyond the hardware budget of basic, low-cost RFID devices such as Electronic Product-Code (EPC) tags [12] whose target cost is in the vicinity of five (U.S.) cents [29]. Higher-cost wireless devices, however, are capable of performing public-key operations, as demonstrated even for the passive RFID tags in e-passports (see, e.g. Nguyen's work [25] for details) and in contactless smartcards. For important personal devices like human-implantable RFID tags [10]—the main targets of our work—we believe that the cost required for the cryptographic circuitry needed in our scheme is within the realm of practicality.

2 Previous Work

2.1 Covert channels

Gus Simmons' pioneering work [31–33] is the foundation for contemporary study of covert channels (sometimes called *subliminal channels*). Cachin [5] first formalized the notion of covert channels in an information-theoretic setting, while Hopper et al. [11] defined the notion in a complexity-theoretic context.

It should be noted that our definition of a covert channel in this paper differs slightly from these previous ones. As generally defined in the literature, a covert channel is a channel that is not feasibly detectable by a "warden," i.e., a polynomial-time observer. We define a covert channel in a somewhat broader sense. In our definition, a covert channel is one that allows the transmission of any data other than tag identifiers. For our purposes, the feasibility of detection is immaterial: Any side information is problematic since it is not explicit in an identification system and can therefore undermine privacy. This definitional distinction is largely a technical one: our CCF-checkable constructions aim to detect any entropy in tag outputs and consequently suppress covert channels as traditionally defined.

Furthermore, while previous models address systems with a single transmitting entity, our definitions must encompass multiple transmitting entities since identifier privacy is only meaningful in a system with multiple transmitting entities.

Young and Yung designed a number of methods to implant covert channels in the key-generation routines of well known public-key cryptosystems [34, 35]. Known as "kleptography," their attacks exploit the probabilistic nature of key generation to transmit private-key data secretly to an attacker via public keys. Their subverted key generation implementations are polynomially indistinguishable from honest ones. Juels and Guajardo [16] propose schemes to suppress kleptographic attacks; their techniques enable verification of covert-freeness in RSA key generation. They use zero-knowledge techniques to prove constraints on the selection of primes.

Choi et al. [8] consider the problem of covert channels in digital signature algorithms—RSA and DSA in particular. They show how to render such algorithms deterministic, thereby permitting verification of covert-freeness. In particular, their constructions replace the random elements of signatures with verifiable, pseudo-random ones. The same authors also proposed constructions for verifying covert-freeness in mix networks [7]. As mix networks are a primitive employed in electronic-election systems, such verification can be important in verifying the preservation of ballot secrecy.

The Choi et al. constructions prove covert-freeness by demonstrating consistency across sequences of values (signatures and mix-network operations in particular), whereas the Juels-Guajardo construction proves covert-freeness of independently generated keys. The challenge we confront in this paper is the need to preserve privacy across sequences of tag outputs while still permitting verification of covert-freeness. That is, we wish to enable checks of consistency on a local basis, e.g. in a pairwise manner, while preserving unlinkability across the sequence of outputs of tags. This unusual tension between linkability and unlinkability — absent in previous research problems in covert-freeness — proved to be a challenging design constraint.

Lepinski et al. [21] demonstrate techniques to suppress covert channels in general multi-party computations. They show in particular how to achieve collusion-free protocols for all finite function computations with publicly observable actions. Their constructions assume that protocol participants are securely initialized by a trusted party. In our case of identification systems, we must relax this assumption because the entity that initializes tags is not necessarily the same entity that wishes to verify the absence of covert channels. Furthermore, the technical apparatus of Lepinski et al. does not naturally accommodate privacy-preserving systems where identities of participants are not readily available as is the case in a privacy-preserving system.

2.2 RFID

As explained above, to achieve simultaneous privacy and CCF-checkability, we loosen the common cryptographic models for identifier privacy. For this purpose, we use a variant on the model proposed for the "minimalist" security protocol of Juels [13]. That model captures the requirement for the physical proximity of an adversary in an RFID system. It assumes that the adversary has only sporadic access to a tag and to legitimate reader-tag authentication sessions. We adopt a similar assumption in our own model, where privacy depends upon "gaps" in an adversary's access to a tag. On the other hand, the independent keying of tags in our system yields a simpler overall model for privacy than the general one for RFID systems recently proposed by Juels and Weis [19].

Our work also draws on the idea of using external, potentially high-power, devices to enforce privacy properties on lightweight devices, an approach employed in public-key-oriented RFID protocols such as [9] and in RFID "firewalls" [18, 27]. Particularly relevant is the work of Ateniese et al. [1], who also use bilinear maps to resolve the tension between privacy and the ability of third parties to verify the authenticity of signatures carried on RFID tags.

A number of papers [30] use symmetric-key cryptography to enforce privacy in the resource-constrained environments typical of RFID systems. Indeed, some researchers have sought to achieve RFID privacy while avoiding on-tag cryptography through techniques such as interference with low-layer RFID protocols [17] or supervision of applications via trusted computing [22]. As explained above, however, we focus on higher-powered RFID tags and wireless devices than those low-power tags for which such symmetric-key schemes were designed.

We again emphasize that public verifiability is an important element of our proposed scheme as it alleviates the key management problem. This problem is important for RFID tags and other lightweight wireless devices because they lack substantive user interfaces. Moreover, as highlighted in libraries [24] and other settings [15, 23], RFID tags undergo frequent changes of possession, exacerbating key mangement problems.

In this paper, we focus exclusively on covert channels in the logical layer of RFID and other wireless systems. Note that, as explained by Avoine and Oechslin [3] (and exploited to extract keys [26]), RFID systems include lower protocol layers where privacy protection also needs to be considered.

For general overviews of RFID privacy (and security), the reader is referred to several surveys [14, 27], as well as the comprehensive online bibliography maintained by Avoine [2].

3 Technical Preliminaries

3.1 Definition of a Tag Identification System

We restrict our investigation here to output-only tags, i.e., tags that emit (privacy-enhanced) identifiers in response to interrogation by readers. Such tags may maintain state; we may assume without loss of generality that this state assumes the form of a simple counter initialized to 0.

Let τ be a security parameter. We say that a (probability) function $\epsilon(\tau) : N \rightarrow [0,1]$ is *negligible* if for every constant c, it is the case that $\epsilon(\tau)$ is asymptotically less than $1/\tau^c$; otherwise, $\epsilon(n)$ is *non-negligible*. Correspondingly, a probability is *overwhelming* if it is equal to $1 - \epsilon(\tau)$ for some negligible quantity $\epsilon(\tau)$. We say that an algorithm is *polynomial time* if its worst-case running time is polynomial in τ.

The system comprises a reader $\mathcal{R}$ and a set of n tags where n is polynomial in τ. The system also includes the following set of polynomial time algorithms:

- $\mathsf{ParmGen}(\tau) \rightarrow (parm, MK)$: This algorithm takes in a security parameter τ and returns the public parameters $parm$ as well as the master key MK for the system.
- $\mathsf{KeyGen}(parm, MK) \rightarrow x$: This algorithm takes as input the public parameters $parm$, along with the master key. It outputs a key x to be programmed into a tag.
- $\mathsf{TagOutput}(c_i, parm, x_i, b) \rightarrow a$: This algorithm determines the output of the tag keyed with x_i on counter value c_i. On calling $\mathsf{TagOutput}$, a tag increments its counter. The value b is an input bit whose function will become clear below.
- $\mathsf{Identify}(parm, MK, a) \rightarrow i$: This algorithm determines the identity of an interrogated tag given its output a. If the tag is not valid, then $\mathsf{Identify}$ outputs the symbol ϕ. Otherwise $\mathsf{Identify}$ outputs an identifier $i \in \{1, 2, \ldots, n\}$. This algorithm is typically executed by the reader.

For notational convenience, we also define an algorithm $\mathsf{KeysGen}(parm, MK, n) \rightarrow (X = \{x_i\}_{i=1}^n)$ that runs KeyGen n times to output n tag keys.

Definition 1. *A tag-identification system* $\mathsf{TIS} = \{\mathsf{ParmGen}, \mathsf{KeyGen}, \mathsf{TagOutput}, \mathsf{Identify}\}$. *A valid system* TIS *is one in which for any* x_i *and any* c_i, $\mathsf{Identify}(parm, MK, \mathsf{TagOutput}(c_i, parm, x_i, b)) \in \{1, \ldots n\}$.

3.2 Covert channel

‘ A *covert channel* for a valid tag-identification system TIS is a mechanism that enables tags to transmit a piece (bit) of information b in addition to tag identifiers. Without loss of generality, we assume that the entity receiving transmissions on the covert channel has access to the key sets X and MK. We assume that the bit b is determined after key generation (otherwise, the keys themselves could encode b).

Let $\mathsf{TagOutput}_i(\cdot, parm, b)$ denote an oracle for tag i; in other words, the output of $\mathsf{TagOutput}_i(c, parm, b)$ is $\mathsf{TagOutput}(c, parm, x_i, b)$.

Definition 2. *A valid* TIS *with parameters* (n, τ) *contains a* covert channel *if there exists some public parameter* $parm$ *and some polynomial time algorithm* $\mathcal{A}$ *such that*

$$
\begin{aligned}
&Adv[\mathcal{A}, \mathsf{TIS}](\tau) = \\
&\quad \Big| \mathsf{Pr}[\mathcal{A}^{\mathsf{TagOutput}_i(\cdot, parm, b)}(X, MK) = b \mid \\
&\qquad X \leftarrow \mathsf{KeysGen}(parm, MK, n)\ ; \\
&\qquad MK \leftarrow \mathsf{ParmGen}(\tau)\ ;\ b \overset{R}{\leftarrow} \{0,1\}] \ - \ 1/2 \Big|
\end{aligned}
$$

is a non-negligible function of τ. We say that a TIS *has no covert channels if there does not exist any polynomial time algorithm that has non-negligible $Adv[\mathcal{A}, \mathsf{TIS}](\tau)$.*

It is easy to see that for any TIS with a probabilistic TagOutput algorithm (with non-negligible entropy), there exists an algorithm $\mathsf{TagOutput}'$ such that $\mathsf{TIS}' = (\mathsf{ParmGen}, \mathsf{KeyGen}, \mathsf{TagOutput}', \mathsf{Identify})$ has a covert channel.

An algorithm that takes a randomized input allows an attacker to replace the input with her own, non-random value. In practice, this tampering is difficult to detect by legitimate users of the system, even with the ability to periodically audit the output of a random-number generator. A system that offers only the ability to choose between two values—as modeled in our system—provides a covert channel for a single bit.

3.3 Privacy

Let x_j and x_k be the private keys for tags j and k. Informally, we say that a tag-identification system TIS is *private* if an entity that does not know the master key MK, the keys x_j, or x_k "cannot" distinguish between the outputs of tags j and k.

Given a TIS and a bit t, we define a *left-right* oracle $\mathsf{LR}[t, c, j, k]$ in the following manner. The oracle outputs $\mathsf{TagOutput}(c, parm, x_j, b)$ if $t = 0$ and outputs $\mathsf{TagOutput}(c, parm, x_k, b)$ if $t = 1$; it outputs ϕ if it was previously queried on either $\mathsf{TagOutput}(c, parm, x_j, \cdot)$ or $\mathsf{TagOutput}(c, parm, x_k, \cdot)$.

Definition 3. *We say that a* TIS *with parameters (n, τ) is private if for all public parameters parm and all polynomial time algorithms $\mathcal{A}$, we have*

$$\Big|\mathsf{Pr}\big[\mathcal{A}^{\mathsf{LR}[t,\cdot,j,k]}(X - \{x_j, x_k\}) = t \mid X \leftarrow \mathsf{KeysGen}(parm, MK, n)\ ;\ t \overset{R}{\leftarrow} \{0,1\}\ ;\ MK \leftarrow \mathsf{ParmGen}(\tau)\ ;\ j, k \overset{R}{\leftarrow} \{1, 2, \ldots, n\}\big] - 1/2\Big|$$

is a negligible function of τ for sufficiently large τ.

Our definitions implicitly exclude the case where tags interact with each other. This assumption reflects common devices such as RFID tags where tags normally interact only with a reader (and not with each other).

4 Covert-channel-freeness vs. privacy

For $z \geq 2$, let us define a z-sequence to be a series of z consecutive outputs from a given tag. We first define a TIS system that is publicly covert-channel-free-checkable (*CCF-checkable*).

Definition 4. *A system* TIS *parametrized by (n, τ) is defined as CCF-checkable if there exists an algorithm* $\mathsf{CCF\text{-}Check}(parm, S) \rightarrow \{0, 1\}$ *on z-sequences S with the following two properties:*

1. Completeness: *Let $S_{i,p()}(\tau)$ be defined as the $p(\tau)$-long sequence of outputs of tag i for a polynomial $p(\tau)$; that is, if $p(\tau) = n$, then for some $k \in \mathbb{Z}$, $S_{i,p()}(\tau) = \mathsf{TagOutput}(k, x_j, parm, b), \mathsf{TagOutput}($ $k + 1, x_j, parm, b), \ldots, \mathsf{TagOutput}(k + n - 1, x_j,$ $parm, b)$. If* TIS *contains a covert channel, then for $b \in \{0, 1\}$, there exists a polynomial p and an $i \in 1, ...n$ such that $S_{i,p()}(\tau)$ contains a z-sequence S with* $\mathsf{CCF\text{-}Check}(Parm, S) = 1$.

2. Soundness: *For any system* TIS *with no covert channel and for any poly-time algorithm* $\mathcal{A}$*, it is the case that*

$$\mathsf{Pr}[\mathsf{CCF\text{-}Check}(parm, S) = 1 \mid b \stackrel{R}{\leftarrow} \{0,1\}\ ; \\ X \leftarrow \mathsf{KeysGen}(parm, MK, n)\ ; \\ S \leftarrow \mathcal{A}^{\mathsf{TagOutput}_i(\cdot, parm, b)}\ ;\ MK \leftarrow \mathsf{ParmGen}(\tau)]$$

is a negligible function of τ *for sufficiently large* τ*.*

Intuitively, completeness here means that for any covert channel for TIS, it is feasible to find a z-sequence that fails the CCF-Check algorithm. Informally, soundness means that CCF-Check never flags a z-sequence when no covert channel exists. (Of course, our definition of CCF-checkability can be extended to inputs S other than z-sequences.)

As we shall now show, the property of public CCF-checkability is fundamentally at odds with privacy. The intuitive reasoning is as follows: in a private TIS, an algorithm $\mathcal{A}$ without knowledge of x_j and x_k (and MK) cannot distinguish between the outputs of tag j and tag k. Thus, intuitively speaking, there is a hidden degree of freedom in the outputs of these tags. The privacy property of TIS implies that $\mathcal{A}$ is unable to perceive if the two tags are "swapped." That is, if $\mathcal{A}$ observes a sequence of outputs from tag j and is then given an output value B randomly extracted from tag j or tag k, $\mathcal{A}$ cannot determine with probability non-negligibly greater than $1/2$ which tag has output the value B.

By "swapping" tags, it is possible to embed a covert channel in a TIS, detected by an appropriately formulated algorithm $\mathcal{A}$. Therefore, if TIS is publicly CCF-checkable, then TIS is not private. Conversely, if TIS is private, then it cannot be publicly CCF-checkable.

Theorem 1. *A tag-identification system* TIS *cannot be both private and publicly CCF-checkable.*

Proof: Suppose that TIS is CCF-checkable. Then there exists an algorithm CCF-Check that takes as input z-sequences and has the completeness and soundness properties specified above.

We create a general covert channel for TIS and we call this modified system with the covert channel $\overline{\mathsf{TIS}}$. We will use overlines to denote the algorithms associated with $\overline{\mathsf{TIS}}$. First select a random pair of tags (j, k); let $p > z$ be an arbitrary counter value. Let $\tilde{X} = \{\tilde{x}_i\}$ be the keys generated by $\overline{\mathsf{KeysGen}}$ (and let $X = \{x_i\}$ as usual be the keys output by KeysGen). If $b = 0$, then $\overline{\mathsf{TIS}}$ gives outputs identical to TIS. If $b = 1$, however, then tag j is "swapped" into tag k for all counter values above p; that is, $c \leq p$

$$\overline{\mathsf{TagOutput}}(c, parm, \tilde{x}_j, b) = \mathsf{TagOutput}(c, parm, x_j, b),$$

and for $c > p$

$$\overline{\mathsf{TagOutput}}(c, parm, \tilde{x}_j, b) = \mathsf{TagOutput}(c, parm, x_k, b).$$

It is easy to see that $\overline{\mathsf{TIS}}$ is a valid tag identification system, and also that $\overline{\mathsf{TIS}}$ contains a covert channel; An algorithm $\mathcal{A}$ can extract the bit b by examining the outputs of all tags for all counter values up to p and seeing if any tag "swaps" itself with another.

By the completeness property of CCF-Check, there exists a polynomial-time algorithm $\mathcal{A}'$ that with non-negligible probability extracts a z-sequence S such that CCF-Check outputs '1' on input S. By the soundness property (that is, CCF-Check never flags, except with negligible probability, a z-sequence when a covert channel does not exist), by the design of $\overline{\mathsf{TIS}}$, this z-sequence S must encompass, except with negligible probability, the "swapping" of tag j into tag k; that is, the z-sequence must include both $\overline{\mathsf{TagOutput}}(p, parm, x_j, b)$ and $\overline{\mathsf{TagOutput}}(p+1, parm, x_k, b)$.

We now use the algorithm CCF-Check to construct an adversary $\mathcal{A}''$ that breaks the privacy of TIS; that is, $\mathcal{A}''$ correctly guesses the t value of the left-right oracle $\mathsf{LR}[t, \cdot, j, k]$ for some $j, k \in [1, n]$.

The adversary $\mathcal{A}''$ operates as follows: with probability at least $1/n^2$ (polynomial in the security parameter τ), the pair of target tags in the privacy game defined above is (j, k). $\mathcal{A}''$ simply makes a pair

of queries $A = \mathsf{TagOutput}_j(p, b)$ and $B = \mathsf{LR}[t, p+1, j, k]$. If $t = 0$, then the oracle LR yields output from tag j; if $t = 1$, then it yields output from tag k. Therefore if $\mathsf{CCF\text{-}Check}(A, B) = 1$, then $\mathcal{A}''$ outputs '1'; otherwise, $\mathcal{A}''$ outputs a random bit.

By the completeness property, $\mathsf{CCF\text{-}Check}$ will detect a "swap" with non-negligible probability, therefore $\mathcal{A}''$ correctly guesses the bit t with non-negligible advantage.

Theorem 1 demonstrates a fundamental conflict between privacy and CCF-checkability. In the remainder of the paper, we explore ways to resolve this conflict. While we cannot contravene Theorem 1, we can achieve a compromise by relaxing our definition of privacy. Our aim is to achieve privacy that is meaningful in a practical sense—if not as strong as possible—and at the same time allows us to achieve CCF-checkability.

5 A CCF-checkable Scheme

In this section we show how to weaken our definition of privacy in order to achieve CCF-checkability. We consider a model in which a continuous observer (or adversary) can correlate appearances of a given tag, but a sporadic observer cannot. We begin by introducing a restricted notion of tag privacy based on *2-clusters*. A 2-cluster is simply an adjacent pair of tag outputs. We craft a scheme that is private against an adversary that does not observe any 2-clusters emitted by a given tag. We then build on our 2-cluster construction to achieve a CCF-checkable construction with Δ-cluster privacy for arbitrary Δ. Such a system is private against an adversary that does not observe any sequence of Δ successive outputs from a single tag, and thus offers stronger privacy than a 2-cluster-private system.

A CCF-checkable tag identification system consists of the four algorithms for a tag identification system, together with an additional algorithm for verifying covert-channel-freeness (CCF):

$\mathsf{CCF\text{-}Check}(\mathsf{TagOutput}, parm) \to \{0, 1\}$: Output a bit indicating covert-channel-freeness for 2-cluster privacy given as input the tag output and public parameters.

We first describe our construction and then the privacy model, before finally proving that the construction is secure under the privacy model.

5.1 A CCF-checkable construction using BDHS

We briefly review bilinear maps and groups of composite order. We use the notation by Boneh, Goh, and Nissim [4], who first introduced composite bilinear groups.

Let $\mathcal{G}$ be an algorithm called a group generator that takes as input a security parameter $\tau \in \mathbb{Z}$ and outputs a tuple $(p, q, \mathbb{G}, \mathbb{G}_1, e)$ where p, q are two distinct primes, $\mathbb{G}, \mathbb{G}_1$ are two cyclic groups of order $N = pq$, and e is a function $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ satisfying the following properties:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: there exists a $g \in \mathbb{G}$ such that $e(g, g)$ has order n in $\mathbb{G}_1$.

We assume that the group action in $\mathbb{G}$ and $\mathbb{G}_1$, together with the bilinear map are computable in polynomial time (parameterized by τ). We also assume that the description of $\mathbb{G}, \mathbb{G}_1$ includes generators of $\mathbb{G}, \mathbb{G}_1$ respectively. We use $\mathbb{G}_p$ to denote the subgroup of order p and similarly for $\mathbb{G}_q$.

Our CCF-checkable construction consists of the following algorithms:

$\mathsf{ParmGen}(\tau)$: Given the security parameter τ as input, run the group generator to obtain the tuple $\mathcal{G}(\tau) = (p, q, \mathbb{G}, \mathbb{G}_1, e)$, where $\mathbb{G}, \mathbb{G}_1$ bilinear groups of composite order $N = pq$, p, q are two primes, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is the bilinear map.

Next find a generator $g \in \mathbb{G}_q$, and find two generators $h_1, h_2 \in \mathbb{G}_p$. Finally, generate $f(\tau) = \ell$, which is the maximum number of queries that a tag will respond to, for some polynomial function $f : \mathbb{Z} \to \mathbb{Z}$. Output $g \in \mathbb{G}_q$ as the private parameters and $parm = (\ell, \mathbb{G}, \mathbb{G}_1, e, h_1, h_2)$ as the public parameters.

$\mathsf{KeyGen}(parm, g)$: Generate a unique random tag identity $d \xleftarrow{R} \mathbb{Z}_n$ and a fixed $r \xleftarrow{R} \mathbb{Z}_n$, then compute the identifier $g^d \in \mathbb{G}$. Output (g^d, r) as the tag private key.

$\mathsf{TagOutput}(i, parm, x)$: On the ith query, a tag with private key $x = (g^d, r)$ emits $A_i = (g^d h_1^{r^{a_i}}, h_2^{r^{a_i}}) \in \mathbb{G} \times \mathbb{G}$ where $a_i = 2a_{i-1}$ and $a_0 = 1$.

$\mathsf{Identify}(parm, g, A_i[0])$: For a reader to identify a tag, it first queries the tag to obtain A_i (assuming this query is the ith query). For a valid response A_i, the computation $e(A_i[0], g) = e(g^d h_1^{r^{a_i}}, g) = e(g^d, g) = e(g, g)^d$ gives the identity of the tag.

$\mathsf{CCF\text{-}Check}(A_i, A_{i+1}, parm)$: To verify covert-channel-freeness for 2-cluster privacy, anyone obtaining two consecutive responses from a tag can perform the following two comparisons:

$$e(A_i[0], A_i[1]) \stackrel{?}{=} e(A_{i+1}[0], h_2) \quad \text{and}$$
$$e(A_i[1], A_i[1]) \stackrel{?}{=} e(A_{i+1}[1], h_2),$$

where $A_i[0]$ denotes the first element in the tuple A_i. Output 1 for a valid check and 0 otherwise. This comparison gives equality for a valid tag because $e(g, h_2) = 1 \in \mathbb{G}_1$ and

$$\begin{aligned} e(A_i[0], A_i[1]) &= e(g^d h_1^{r^{a_i}}, h_2^{r^{a_i}}) \\ &= e(g^d, h_2^{r^{a_i}}) \cdot e(h_1^{r^{a_i}}, h_2^{r^{a_i}}) \\ &= e(h_1^{r^{a_{i+1}}}, h_2) \\ &= e(g^d h_1^{r^{a_{i+1}}}, h_2). \end{aligned}$$

The output of the $\mathsf{Identify}$ algorithm $e(g, g)^d$ is sufficient to identify the tag using a lookup table of identities. Consequently, the reader need only keep a record per tag as opposed to a record per chip/per sample as in some naive schemes. Furthermore, note that required value for ℓ is an upper bound on the number of times a tag is expected to be queried for an output. For example, a tag that provides access into a building need only have $\ell = 91250$ to have a lifetime of 5 years assuming 50 accesses a day. We next describe the security model for a CCF-checkable TIS and then prove the security of our construction in this model.

Security for 2-cluster Privacy Our security model aims to capture the notion that a continuous observer can correlate appearances of a given tag, but a sporadic observer cannot. Roughly speaking, a 2-cluster CCF-checkable sequence $\mathcal{A} = A_1, A_2, \ldots$ emitted by a tag has two properties: (1) An adversary cannot feasibly link elements in any subsequence that contains no two elements in succession, i.e., no pair (A_i, A_{i+1}), and conversely, (2) a verifier can check the correctness of any subsequence of length 2, i.e., any pair. In section 5.2, we show how to convert a TIS with 2-cluster privacy into one with Δ-cluster privacy for $\Delta > 2$. For now, we only consider 2-cluster sequences.

Define the following two experiments for a 2-cluster CCF-checkable tag identification scheme $\mathcal{F}$:

Experiment b $(b = 0, 1)$

- The challenger $\mathcal{C}$ generates public parameters using $\mathsf{ParmGen}(\tau)$, and gives the public parameters $(\ell, \mathbb{G}, \mathbb{G}_1, e, h_1, h_2)$ to the adversary $\mathcal{A}$. Next, for two tags d_0, d_1, $\mathcal{C}$ runs KeyGen on to obtain their private keys x_0, x_1.

– $\mathcal{A}$ then makes a series of queries, each of which is either a test query or a challenge query. $\mathcal{A}$ is allowed to make at any time and in any order, any (polynomial) number of test queries, but only Q_c challenge queries. Note that ℓ is the maximum index given by $parm$.

  - $\mathcal{A}$ can obtain outputs from either tag at any index and in any order using test queries. On receiving a test query $(t_i, \gamma) \in ([1, \ell] \times \{0, 1\})$, $\mathcal{C}$ runs $\mathsf{TagOutput}(t_i, parm, x_\gamma)$ and returns the response to $\mathcal{A}$.
  - $\mathcal{A}$ can specify the index on which she wants to challenged on by issuing challenge queries (with a restriction described later). On receiving a challenge query $c_i \in [1, \ell]$, $\mathcal{C}$ runs $\mathsf{TagOutput}(c_i, parm, x_b)$ and returns the response to $\mathcal{A}$.

    Roughly speaking, the restriction on the challenge queries is that no challenge query index can be adjacent to a test query index (but challenge query indexes can be adjacent to each other, and similarly for test query indexes).

    Denote the number of test queries by Q_t. More precisely, let $T = \{t_1, t_2, \ldots, t_{(Q_t)}\}$ and let $C = \{c_1, c_2, \ldots, c_{(Q_c)}\}$; the restriction on challenge queries is that $C \cap T = (C \pm 1) \cap T = \emptyset$ where $C + 1$ denotes adding $1 \in \mathbb{Z}$ to every member of C.

– At the end of the game, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

If W_b is the event that $\mathcal{A}$ outputs 1 in Experiment b, we define $\mathcal{A}$'s advantage with respect to $\mathcal{F}$ as

$$\mathrm{Adv}[\mathcal{A}_{(\ell,Q_c)}, \mathcal{F}] = |\Pr[W_0] - \Pr[W_1]| .$$

where the probability is over the random bits used by the challenger and the adversary.

Definition 5. *A 2-cluster CCF-checkable* TIS *scheme $\mathcal{F}$ is (ℓ, Q_c)-secure if for all poly-time adversaries $\mathcal{A}$ that make Q_c challenge queries, $Adv[\mathcal{A}_{(\ell,Q_c)}, \mathcal{F}] < \epsilon$ where ϵ is a negligible function of τ for sufficiently large τ. We say that $\mathcal{A}$ is a (ℓ, Q_c)-adversary if $\mathcal{A}$ breaks the (ℓ, Q_c)-secure CCF scheme with advantage ϵ where ϵ is a non-negligible function in τ.*

In the full version of the paper, we describe a more general security model involving n tags instead of just 2 tags. Summarizing the model briefly, in the setup phase, the challenger picks n tags and randomly selects one of them to be the challenge tag (with which he uses to answer challenge queries). In the course of the game, the adversary performs the usual test and challenge queries, as well as at most $n - 2$ reveal queries. A reveal query on a tag forces the challenger to disclose the secret keys for that particular tag. There are restrictions on the choice of challenge and test queries similar to those detailed in the model above. The rest of the model is identical to the one laid out above.

If the tag keys of a CCF-checkable construction are generated independently at random, then it is easy to show that if the CCF-checkable construction is secure in a game with 2 tags and no reveal queries, then it is also secure using a game with n tags and $n - 2$ reveal queries (modulo a $1/(n - 2)$ decrease in security). Since our construction has tag keys that are chosen independently at random, in the interests of clarity, we prove our scheme under the slightly weaker security model stated above.

Security Proof We prove that our CCF-checkable construction is secure using the Bilinear Diffie-Hellman Squaring that we now define. For a given group generator $\mathcal{G}$, define the following distribution

$D(\tau)$:

$$
\begin{aligned}
&\ell\text{-BDHS}: \\
&(p, q, \mathbb{G}, \mathbb{G}_1, e) \xleftarrow{R} \mathcal{G}(\tau), \\
&N \leftarrow pq, \;\; v \xleftarrow{R} \mathbb{G}_q, \;\; w \xleftarrow{R} \mathbb{G}_p, \;\; \beta \xleftarrow{R} \mathbb{Z}_p, \\
&Z \leftarrow \Big((N, \mathbb{G}, \mathbb{G}_1, e), v, w, w^{\beta}, w^{(\beta^2)}, w^{(\beta^4)}, \ldots, \\
&\qquad w^{(\beta^{(2^{\ell-2})})}, w^{(\beta^{(2^{\ell+2})})}, w^{(\beta^{(2^{\ell+3})})}, \ldots, w^{(\beta^{(2^{2\ell})})}\Big), \\
&Q \leftarrow w^{(\beta^{(2^{\ell})})}; \\
&\text{Output } (Z, Q)
\end{aligned}
$$

For an algorithm $\mathcal{A}$, define $\mathcal{A}$'s advantage in solving the composite ℓ-BDHS for $\mathcal{G}$ as

$$
\begin{aligned}
&\text{Adv}[\mathcal{A},\ \mathcal{G},\ \ell\text{-BDHS}](\tau) = \\
&\qquad \left|\Pr[\mathcal{A}(Z, Q) = 1] - \Pr[\mathcal{A}(Z, R) = 1]\right|,
\end{aligned}
$$

where $(Z, Q) \xleftarrow{R} D(\tau)$ and $R \xleftarrow{R} \mathbb{G}_p$.

Definition 6. *We say that $\mathcal{G}$ satisfies the composite ℓ-BDHS assumption if for any polynomial time algorithm $\mathcal{A}$, we have that $Adv[\,\mathcal{A},\ \mathcal{G},\ \ell\text{-}BDHS](\tau) < \epsilon$ is a negligible function of τ for sufficiently large τ.*

Theorem 2. *If the 2ℓ-BDHS assumption holds in $\mathcal{G}$, then the CCF construction is $(\ell,\ Q_c)$-secure for any $Q_c < \ell/3$.*

Proof of Theorem 2: We prove the theorem in two steps: 1) We first show that an adversary that makes Q_c challenge queries and has non-negligible advantage in the security game can be used to build another adversary that only makes one challenge query but also has non-negligible advantage. 2) We prove that our CCF construction is secure by arguing the contrapositive and using the lemma from the first part.

Step 1. We start part 1 of the proof of Theorem 2 with the following lemma.

Lemma 1. *For any (ℓ, Q_c)-adversary $\mathcal{A}$ with a polynomial running time that breaks the CCF scheme with advantage ϵ, there exists a $(\ell, 1)$-adversary $\mathcal{B}$ with a polynomial running time that breaks the same scheme with advantage ϵ/Q_c.*

Proof: Let Q_t denote the number of test queries $\mathcal{A}$ makes. We first show via a hybrid argument on a series of experiments that $\mathcal{A}$ has non-negligible advantage in distinguishing a specific challenge returned in response to one of $\mathcal{A}$'s challenge queries. Define a series of hybrid experiments as follows:

1. Experiment H_0 = Experiment 0 (as defined in the security model)
2. Experiment $H_{(Q_c-1)}$ = Experiment 1 (as defined in the security model)
3. Experiment H_i where $i \in [1, Q_c - 2]$ is defined exactly as Experiment 0, except for the challenger $\mathcal{C}$'s responses to the challenge queries; the first i challenge replies are generated by $\mathcal{C}$ using (g^{d_0}, r_0) (tag d_0's private keys) as input to the TagOutput algorithm, whereas the subsequent $Q_c - i$ challenge replies are generated by $\mathcal{C}$ using (g^{d_1}, r_1) (tag d_1's private keys) as input to the TagOutput algorithm.

A standard hybrid argument shows that A has advantage at least ϵ/Q_c in distinguishing between Experiment H_j and Experiment H_{j+1} for some $j \in [0, Q_c - 1]$.

We will use $\mathcal{A}$ (which breaks the (ℓ, Q_c)-secure TIS with advantage ϵ) to build an algorithm $\mathcal{B}$ that has advantage at least ϵ/Q_c in breaking a $(\ell, 1)$-secure TIS. $\mathcal{B}$ interacts with a challenger $\mathcal{C}$ and $\mathcal{A}$ as follows.

$\mathcal{B}$ obtains the public parameters from $\mathcal{C}$, and passes them to $\mathcal{A}$. When $\mathcal{A}$ issues test queries to $\mathcal{B}$, $\mathcal{B}$ passes them along to $\mathcal{C}$ and returns the replies from $\mathcal{C}$ directly to $\mathcal{A}$. For each of the first j challenge queries $\{c_1, \ldots, c_{j-1}\}$ that $\mathcal{A}$ issues to $\mathcal{B}$, $\mathcal{B}$ issues a test query $(c_i, 0)$ to $\mathcal{C}$ where $c_i \in \{c_1, \ldots, c_{j-1}\}$, and gives the reply from $\mathcal{C}$ to $\mathcal{A}$. For the jth challenge query c_j from $\mathcal{A}$ to $\mathcal{B}$, $\mathcal{B}$ issues it as a challenge query to $\mathcal{C}$ and gives the reply to $\mathcal{A}$. For each of subsequent $Q_c - j - 1$ challenge queries $\{c_{j+1}, \ldots, c_{(Q_c)}\}$ that $\mathcal{A}$ issues to $\mathcal{B}$, $\mathcal{B}$ issues a test query $(c_i, 1)$ to $\mathcal{C}$ where $c_i \in \{c_{j+1}, \ldots, c_{(Q_c)}\}$, and gives the reply from $\mathcal{C}$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs a bit b', which $\mathcal{B}$ uses as its output.

From its interaction with $\mathcal{B}$, $\mathcal{A}$'s view of the simulation is either that of Experiment H_j or H_{j+1}. We showed above that $\mathcal{A}$ has advantage ϵ/Q_c in distinguishing between Experiment H_j and Experiment H_{j+1}. Note that $\mathcal{B}$ makes exactly $Q_t + Q_c - 1$ test queries and 1 challenge query to $\mathcal{C}$, and has approximately the same running time as $\mathcal{A}$. Therefore, $\mathcal{B}$ also has the same non-negligible advantage $\mathrm{Adv}[\mathcal{A}_{\ell,1}, \mathcal{CCF}] \geq \epsilon/Q_c$ in breaking the scheme. □

Step 2: We now begin the second part of the proof of Theorem 2. Suppose the CCF construction is not (ℓ, Q_c)-secure, then there exists a (ℓ, Q_c)-adversary $\mathcal{A}'$ that has advantage ϵ in breaking the CCF construction. By Lemma 1, there also exists a $(\ell, 1)$-adversary $\mathcal{A}$ that breaks the CCF scheme with advantage ϵ/Q_c while making $Q_t + Q_c - 1$ test queries and 1 challenge query. We describe an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the $(2\ell, \epsilon)$-BDHS assumption in $\mathbb{G}$.

Define $W_i = w^{\beta^{(2^i)}}$ where $i \in [0, 4\ell]$. $\mathcal{B}$ is given $Z \leftarrow (N, \mathbb{G}, \mathbb{G}_1, \epsilon, v, w, \boldsymbol{W})$ where $\boldsymbol{W} = (W_0, W_1, \ldots, W_{2\ell-2}, W_{2\ell+2}, W_{\ell+3}, \ldots, W_{4\ell})$ and S. We will show that $\mathcal{B}$ has non-negligible advantage in deciding if $W_{2\ell} = S \stackrel{?}{=} w^{(\beta^{(2^{2\ell})})}$.

$\mathcal{B}$ first flips a coin $b \in \{0, 1\}$ to decide which experiment to simulate. $\mathcal{B}$ generates the public parameters of the CCF system: $\mathcal{B}$ first sets $g = v \in \mathbb{G}_q$. $\mathcal{B}$ picks a random $a \in [0, N-1]$ and creates another generator $w_1 \in \mathbb{G}_p$ by setting $w_1 = w^a$. If $w_1 = 1$, then $\mathcal{B}$ can factor N and breaks the assumption; this event happens with probability $1/N$. We assume $\mathcal{B}$ has not factored N. For notational convenience, we define $w_0 = w$. $\mathcal{B}$ gives the public parameters $parm = (\ell, \mathbb{G}, \mathbb{G}_1, e, w_0, w_1)$ as input to $\mathcal{A}$. Note that since w_0, w_1 are uniformly distributed generators in $\mathbb{G}_p$, the simulated public parameters given to $\mathcal{A}$ are from the same distribution as public parameters generated using ParmGen.

Before answering any queries, $\mathcal{B}$ picks $d_0, d_1 \stackrel{R}{\leftarrow} [0, N-1]$ to create g^{d_0} and g^{d_1}. In addition, $\mathcal{B}$ first picks a random $s \in [0, 2\ell - 1]$. Roughly speaking, $\mathcal{B}$'s choice of s fixes the start of a contiguous interval of size 2ℓ from $\boldsymbol{W}$ with which $\mathcal{B}$ uses to answer queries; note that any choice of this interval contains S. $\mathcal{B}$ also picks a $\gamma \in [0, N-1]$ with which he will re-randomize the W_i terms to answer tag d_1 test queries. When $\mathcal{A}$ issues test and challenge queries, $\mathcal{B}$ first sets $bad \leftarrow 0$ and then answers them in the following fashion:

- Test Queries: On receiving $(t_i, \theta) \in ([1, \ell] \times \{0, 1\})$ from $\mathcal{A}$, $\mathcal{B}$ simulates the reply as follows:
  - For $\theta = 0$, $\mathcal{B}$ replies with
    $$(g^{d_0} \cdot W_{s+t_i}\ ,\ (W_{s+t_i})^a).$$
    If $s + t_i \in [2\ell - 1, 2\ell + 1]$, $\mathcal{B}$ sets $bad \leftarrow 1$.
  - For $\theta = 1$, $\mathcal{B}$ replies with
    $$(g^{d_1} \cdot (W_{s+\ell+t_i})^\gamma\ ,\ (W_{s+\ell+t_i})^{a\gamma}).$$
    If $s + \ell + t_i \in [2\ell - 1, 2\ell + 1]$, $\mathcal{B}$ sets $bad \leftarrow 1$.

  In both cases, an easy computation shows that $\mathcal{B}$'s simulated reply (if $\mathcal{B}$ does not set $bad \leftarrow 1$) comes from the same distribution as an actual reply. Also, since query indexes are from the interval $[1, \ell]$, $\mathcal{B}$ can answer all test queries apart from the three queries listed for $\theta = 0, 1$ (which result in $bad = 1$).

  If $bad = 1$, then $\mathcal{B}$ outputs a uniformly random bit and ends the simulation.

– Challenge Query: On receiving c_i, if $b = 0$ and $c_i \neq 2\ell$, $\mathcal{B}$ sets $bad \leftarrow 1$; if $b = 1$ and $c_i \neq 2\ell + s$, $\mathcal{B}$ sets $bad \leftarrow 1$. Otherwise, $\mathcal{B}$ replies with $(g^{d_1}(W_{2\ell})^{\gamma}, (W_{2\ell})^{a\gamma})$ for experiment 1 and $(g^{d_0}W_{2\ell}, (W_{2\ell})^{a})$ for experiment 0.
If $bad = 1$, then $\mathcal{B}$ outputs a uniformly random bit and ends the simulation.

If $bad = 0$ throughout the simulation, $\mathcal{A}$ eventually finishes her queries and then outputs a bit b'. If $b' = b$, $\mathcal{B}$ outputs 1; otherwise $\mathcal{B}$ outputs 0.

We now analyze $\mathcal{B}$'s advantage.

– If $bad = 0$ and $S = w^{(\beta^{(2^{2\ell})})}$, then $\mathcal{B}$'s reply to $\mathcal{A}$'s challenge query is from the same distribution as a real reply, and $\mathcal{A}$ has advantage ϵ/Q_c in determining b.
– If $bad = 0$ and $S \stackrel{\mathrm{R}}{\leftarrow} \mathbb{G}_p$, then $\mathcal{B}$'s reply to the challenge query is from the uniform distribution on $\mathbb{G}_p$, and $\mathcal{A}$ has no advantage in determining b.
– If $bad = 1$ and $\mathcal{B}$ ends the simulation early, then $\mathcal{B}$ outputs a random bit, implying that $\mathcal{B}$ has no advantage in solving the 2ℓ-BDHS problem.
– Since s is chosen uniformly at random from $[0, 2\ell-1]$, $W_{2\ell} = S$ is located uniformly at random within the interval of $\{W_s, W_{s+1}, \ldots, W_{s+2\ell}\}$ from which $\mathcal{B}$ answers queries. Therefore, the probability of $\mathcal{A}$ issuing a challenge query on $W_{2\ell}$ is $1/2\ell$, and this event is independent from the event that $\mathcal{A}$ outputs 1 (or 0).

From this analysis, we can compute that the advantage of $\mathcal{B}$ in breaking the 2ℓ-BDHS assumption is $\epsilon/Q_c \cdot 2\ell$, which is non-negligible. □

5.2 Δ-Cluster Sequences

We now briefly show that given a 2-cluster sequence $\mathcal{A} = A_1, A_2, \ldots$, we can derive a Δ-cluster sequence $\mathcal{A}' = A'_1, A'_2, \ldots$. Note that the 2-cluster property of $\mathcal{A}$ implies that the "staggered" subsequence $A_1, A_3, A_5 \ldots$ of $\mathcal{A}$ is private. Thus to construct a Δ-cluster sequence $\mathcal{A}'$, we can start with a staggered subsequence in $\mathcal{A}$ and embellish it with ciphertexts on the missing elements (i.e., $A_2, A_4, \ldots$). The decryption key for any of these ciphertexts is a subsequence of length Δ in $\mathcal{A}$. Thus, knowledge of a Δ-cluster permits "filling in" of a portion of the underlying staggered subsequence and hence use of the 2-cluster property of $\mathcal{A}$ to check correctness.

Let $|\mathbb{G}|$ denote the bit size of the representation of the group $\mathbb{G}$. For our construction, we require an adaptive chosen-ciphertext secure symmetric cipher E and an $|\mathbb{G}|$-exposure resilient function [6] (ERF) $f : (\mathbb{G} \times \mathbb{G})^{\Delta} \rightarrow \{0,1\}^l$ on Δ tag elements. Roughly speaking, an $|\mathbb{G}|$-ERF is a function such that if an adversary learns all but $|\mathbb{G}|$ bits of the input, then the output of the function still appears pseudo-random to the adversary. We construct $\mathcal{A}'$ in three stages:

1. We embed the staggered subsequence in $\mathcal{A}'$, that is, we let $A'_i = A_{2i-1}$ for all i.
2. We construct ciphertexts of the missing elements of the staggered subsequence. That is, let $k_i = f(A'_i, A'_{i+1}, \ldots, A'_{i+\Delta-1})$ be a key derived from a given Δ-cluster in $\mathcal{A}'$. We let $C_i = E_{k_i}[A_{2i}]$, where E_k denotes encryption under key k in a suitable symmetric-key cipher.
3. For all i, we append C_i to A'_i. That is, we let $A'_i = (A_{2i-1}, C_i)$.

Given any Δ-cluster $A'_i, A'_{i+1}, \ldots, A'_{i+\Delta-1}$, a verifier can decrypt C_i to obtain A_{2i}. Since $A'_i = A_{2i-1}$ and $A'_{i+1} = A_{2i+1}$, it follows that the verifier learns the 3-cluster $A_{2i-1}, A_{2i}, A_{2i+1}$. The verifier can then check the two adjacent pairs of elements in this subsequence, thereby verifying the validity of the pair (A'_i, A'_{i+1}) in $\mathcal{A}'$.

This Δ-cluster construction doubles the size of tag output elements. That is, it imposes a small communication cost that is independent of Δ.

The 2-cluster security model can be extended to Δ-cluster security by modifying the restrictions on challenge queries — instead of $C \cap T = (C \pm 1) \cap T = \emptyset$ for 2-cluster privacy, we require that $C \cap T = (C \pm 1) \cap T = (C \pm 2) \cap T = \ldots = (C \pm \Delta) \cap T = \emptyset$. The proof of security is straightforward and follows directly from the 2-cluster privacy of the sequence emitted by the tag, the exposure resilience of f, and adaptive chosen-ciphertext security on E. (Weaker assumptions on E can suffice, but require careful construction, as the adversary can effectively control tag counters.)

6 Conclusions

We showed that CCF-checkability and identifier privacy cannot co-exist in a single tag identification system under the strongest definition of privacy. We developed a weaker privacy model where a continuous observer can correlate appearances of a given tag, but a sporadic observer cannot. We constructed a privacy-preserving tag identification scheme that is CCF-checkable and proved it secure under the weaker privacy model using the ℓ-BDHS complexity assumption.

References

1. G. Ateniese, J. Camenisch, and B. de Medeiros. Untraceable RFID tags via insubvertible encryption. In *Proceedings of ACM CCS 2005*, pages 92–101, 2005.
2. G. Avoine. Security and privacy in RFID systems. Online bibliography. Referenced 2007 at http://lasecwww.epfl.ch/~gavoine/rfid.
3. G. Avoine and P. Oechslin. RFID traceability: A multilayer problem. In A. Patrick and M. Yung, editors, *Proceedings of Financial Cryptography 2005*, volume 3570 of *LNCS*, pages 125–140. Springer, 2005.
4. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In J. Kilian, editor, *Proceedings of the Theory of Cryptography Conference*, volume 3378 of *LNCS*, pages 325–342, 2005.
5. C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Proceedings of Information Hiding 1998*, volume 1525 of *LNCS*, pages 306–318. Springer, 1998.
6. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-resilient functions and all-or-nothing transforms. In B. Preneel, editor, *Proceedings of Eurocrypt 2000*, volume 1807 of *LNCS*, pages 453–469. Springer, 2000.
7. J. Y. Choi, P. Golle, and M. Jakobsson. Auditable privacy: On tamper-evident mix networks. In G. D. Crescenzo and A. Rubin, editors, *Proceedings of Financial Cryptography 2006*, volume 4107 of *LNCS*, pages 126–141. Springer, 2006.
8. J. Y. Choi, P. Golle, and M. Jakobsson. Tamper-evident digital signatures: Protecting certification authorities against malware. In *Proceedings of the Symposium on Dependable Autonomic and Secure Computing 2006*, pages 37–44. IEEE, 2006.
9. P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets. In T. Okamoto, editor, *RSA Conference - Cryptographers' Track*, volume 2964 of *LNCS*, pages 163–178, 2004.
10. J. Halamka, A. Juels, A. Stubblefield, and J. Westhues. The security implications of VeriChip cloning. *Journal of the American Medical Informatics Association*, 13(6):601–607, 2006.
11. N. Hopper, J. Langford, and L. von Ahn. Provably secure steganography. In M. Yung, editor, *Proceedings of Crypto 2002*, volume 2442 of *LNCS*, pages 77–92. Springer, 2002.
12. E. Inc. Class 1 generation 2 UHF air interface protocol standard version 1.0.9. Referenced 2007 at http://www.epcglobalinc.com/standards_technology/EPCglobalClass-1Generation-2UHFRFIDProtocolV109.pdf.
13. A. Juels. Minimalist cryptography for RFID tags. In *SCN '04*, pages 149–164, 2004.
14. A. Juels. RFID security and privacy: A research survey. *J-SAC*, 24(2):381–394, February 2006.
15. A. Juels, S. Garfinkel, and R. Pappu. RFID privacy: An overview of problems and proposed solutions. *IEEE Security and Privacy*, 3(3):34–43, May/June 2005.

16. A. Juels and J. Guajardo. RSA key generation with verifiable randomness. In D. Naccache and P. Paillier, editors, *Proceedings of PKC 2002*, volume 2274 of *LNCS*, pages 72–86. Springer, 2002.
17. A. Juels, R. Rivest, and M. Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In V. Atluri, editor, *Proceedings of ACM CCS 2003*, pages 103–111. ACM Press, 2003.
18. A. Juels, P. Syverson, and D. Bailey. High-power proxies for enhancing RFID privacy and utility. In G. Danezis and D. Martin, editors, *Privacy Enhancing Technologies (PET)*, 2005.
19. A. Juels and S. Weis. Defining strong privacy for RFID. Cryptology ePrint Archive, Report 2006/137. Short abstract to appear in PerTec '07.
20. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Proceedings of Crypto 1996*, volume 1109 of *LNCS*, pages 104–113. Springer, 1996.
21. M. Lepinski, S. Micali, and A. Shelat. Collusion-free protocols. In H. Gabow and R. Fagin, editors, *Proceedings of STOC 2005*, pages 543–552. ACM, 2005.
22. D. Molnar, A. Soppera, and D. Wagner. Privacy for RFID through trusted computing (short paper). In S. D. C. di Vimercati and R. Dingledine, editors, *Proceedings of WPES*, 2005.
23. D. Molnar, A. Soppera, and D. Wagner. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In B. Preneel and S. Tavares, editors, *SAC '05*, LNCS. Springer, 2005.
24. D. Molnar and D. Wagner. Privacy and security in library RFID : Issues, practices, and architectures. In B. Pfitzmann and P. McDaniel, editors, *Proceedings of ACM CCS 2004*, pages 210 – 219. ACM Press, 2004.
25. K. Nguyen. Elliptic curves and MRTDs. In *Interfest Singapore*, 2005. Slide presentation.
26. Y. Oren and A. Shamir. Power analysis of RFID tags, 2006. Referenced 2006 at http://www.wisdom.weizmann.ac.il/˜yossio/rfid/.
27. M. Rieback, B. Crispo, and A. Tanenbaum. RFID Guardian: A battery-powered mobile device for RFID privacy management. In C. Boyd and J. M. González Nieto, editors, *ACISP '05*, volume 3574 of *LNCS*, pages 184–194. Springer, 2005.
28. S. Saponas, J. Lester, C. Hartung, and T. Kohno. Devices that tell on you: The Nike+iPod sport kit. Technical Report 2006-12-06, University of Washington, 2006.
29. S. Sarma. Towards the five-cent tag. Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001. Available from http://www.epcglobalinc.org.
30. S. E. Sarma, S. A. Weis, and D. Engels. Radio-frequency identification systems. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Workshop on Cryptographic Hardware and Embedded Systems – CHES '02*, volume 2523 of *LNCS*, pages 454–469. Springer, 2002.
31. G. Simmons. The prisoners' problem and the subliminal channel. In D. Chaum, editor, *Proceedings of Crypto 1983*, pages 51–67. Plenum Press, 1983.
32. G. Simmons. The subliminal channel and digital signatures. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Proceedings of Eurocrypt 1984*, volume 0209 of *LNCS*, pages 364–378. Springer, 1984.
33. G. Simmons. Subliminal communication is easy using the DSA. In T. Helleseth, editor, *Proceedings of Eurocrypt 1993*, volume 765 of *LNCS*, pages 218–232. Springer, 1993.
34. A. Young and M. Yung. The dark side of black-box cryptography, or: Should we trust capstone? In N. Koblitz, editor, *Proceedings of Crypto 1996*, volume 1109 of *LNCS*. Springer, 1996.
35. A. Young and M. Yung. Kleptography: Using cryptography against cryptography. In W. Fumy, editor, *Proceedings of Eurocrypt 1993*, volume 1233 of *LNCS*. Springer, 1997.