# Anonymity in Shared Symmetric Key Primitives

Gregory M. Zaverucha and Douglas R. Stinson
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{gzaveruc, dstinson}@uwaterloo.ca

February 4, 2009

**Abstract**

We provide a stronger definition of anonymity in the context of shared symmetric key primitives, and show that existing schemes do not provide this level of anonymity. A new scheme is presented to share symmetric key operations amongst a set of participants according to a $(t, n)$-threshold access structure. We quantify the amount of information the output of the shared operation provides about the group of participants which collaborated to produce it.

## 1 Introduction

In this paper, we consider the anonymity provided by schemes for sharing symmetric key operations such as block ciphers or message authentication codes (MACs). We will focus on threshold access structures. Let $\mathcal{P}$ be a set of participants. A $(t, n)$-*threshold access structure* on $\mathcal{P}$ is defined by $\Gamma$, which is the set of all authorized sets, namely $\Gamma = \{A \subseteq \mathcal{P} : |A| = t\}$. Put simply, any subset of participants of size at least $t$ is authorized. In this model, the keys are distributed by an entity called the *receiver*, who will later receive a message from some $A \in \Gamma$.

One could simply distribute key shares according to a $(t, n)$ secret sharing scheme, however, the participants must actually learn the key in order to use it for a symmetric operation. To avoid this requires an approach which uses the shares directly.

Work on sharing block ciphers was initiated by Brickell et al. [4], using a variant of secret sharing called sequence sharing. Subsequently, improved schemes for distributing block ciphers using cumulative arrays and perfect hash families were studied by Martin et al. in [9]. They introduce generalized cumulative arrays (GCAs) and give some efficient solutions (in the number of keys). A second paper by Martin et al. extends [9] to consider distributing the computation of MACs as well [10].

If a set of participants $A \in \Gamma$ collaborates to encrypt a message, what information does the receiver learn about $A$? Informally, anonymity is provided if the identity of $A$ is kept secret from the receiver. We will define anonymity for groups of participants as well as individual participants in Section 2.

In addition to shared block ciphers and MACs, schemes using generalized cumulative arrays have been used in secret sharing [8] and for sharing pseudo random functions [18]. A GCA is defined below.

**Definition 1.1.** A *generalized cumulative array* (GCA) is a set $Y = \{y_1, \ldots, y_m\}$, a partition of $Y$, say $\{K_1, \ldots, K_v\}$, a set of subsets of $Y$, denoted by $\mathcal{B} = \{B_1, \ldots, B_n\}$, and an integer $t$, such that the following properties are satisfied.

(i) For any $t$-set $A \subseteq \mathcal{B}$, $K_i \in \left(\bigcup_{B \in A} B\right)$ for **at least one** $i \in [1, \ldots, v]$.

(ii) For any set $A' \subseteq \mathcal{B}$ with $|A'| < t$, $K_j \notin \left(\bigcup_{B \in A'} B\right)$ for **all** $j \in [1, \ldots, v]$.

When $|B_i| = \ell$ for all $B_i \in \mathcal{B}$, the GCA is *$\ell$-uniform*. In this paper we consider only $\ell$-uniform GCAs for $(t, n)$-threshold access structures, and adopt the notation $\mathrm{GCA}(t, n; \ell, v)$. In the schemes we consider, $Y$ is a set of keys, and each participant is given one of the sets in $\mathcal{B}$.

Long et al. explore GCAs in the context of secret sharing and give upper and lower bounds for existence of GCAs [8]. Martin and Ng [11] also give constructions of GCAs and metrics to describe the efficiency of GCAs. The best construction for threshold access structures in both papers uses perfect hash families.

**Definition 1.2.** Let $X$ be a set of size $n$, and $Y$ be a set of size $m$. An $(\ell; n, m)$-hash family is a collection $\mathcal{F}$ of $\ell$ functions from $X$ to $Y$. $\mathcal{F}$ is called a *perfect hash family* of *strength* $t$ if for any distinct inputs $c_1, \ldots, c_t \in X$, there exists some $f \in \mathcal{F}$ such that $f(c_i) \neq f(c_j)$ for all $i \neq j$, $1 \leq i, j \leq t$. We use the notation $\mathrm{PHF}(\ell; n, m, t)$.

We will usually depict a $\mathrm{PHF}(\ell; n, m, t)$ as an $\ell \times n$ array populated with $m$ symbols. Each column represents an element $x_j \in X$, and each row is defined by a function. The $(i, j)$-th entry is defined to be $f_i(x_j)$. This array has the property that within any $\ell \times t$ subarray, there is a row containing $t$ distinct elements.

A $\mathrm{PHF}(\ell; n, t, t)$ naturally defines a $\mathrm{GCA}(t, n; \ell, t)$ as follows. Let $Y = \{(i, j) : i \leq \ell, j \leq t\}$, and $K_i = \{(i, 1), \ldots, (i, t)\}$. Each $B_i \in \mathcal{B}$ is a column of the matrix representation of the PHF, along with row indices i.e., $B_k = \{(i, f_i(k)) : 1 \leq i \leq \ell\}$.

Property (i) is satisfied since for any $A = \{B_{i_1}, \ldots, B_{i_t}\}$, there will be some $f$ such that $f(i_1) \neq \ldots \neq f(i_t)$ and hence $A$ will have $t$ distinct values in one row, and therefore contain some $K_i$. However, $A' = \left\{B_{i_1}, \ldots, B_{i_{t-1}}\right\}$ has at most $t-1$ distinct values at each row and therefore cannot cover any $K_i$. Hence property (ii) is satisfied as well.

Long et al. [8] prove that the PHF construction gives asymptotically optimal GCAs. Since practical and efficient constructions for GCAs are only known for

threshold access structures, our anonymity study will focus on threshold GCAs constructed from PHFs as described above. We now give a small example of this construction.

**Example 1.3.** Let $f_1, f_2$ be a PHF$(2; 4, 2, 2)$:

| | | | |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| 1 | 1 | 2 | 2 |

.

The construction described above is used to construct a GCA$(2, 4; 2, 2)$, where $Y = \{(1,1),(1,2),(2,1),(2,2)\}$, $K_1 = \{(1,1),(1,2)\}$, $K_2 = \{(2,1),(2,2)\}$.

| $B_1$ | $B_2$ | $B_3$ | $B_4$ |
|---|---|---|---|
| $(1, f_1(1))$ | $(1, f_1(2))$ | $(1, f_1(3))$ | $(1, f_1(4))$ |
| $(2, f_2(1))$ | $(2, f_2(2))$ | $(2, f_2(3))$ | $(2, f_2(4))$ |

Since the schemes we will present assign $B_i$ from the GCA to $P_i$ for all $P_i \in \mathcal{P}$, it makes sense to talk about a "column of keys" given to $P_i$ and a "row which separates $A \in \Gamma$". If row $r$ separates $A \in \Gamma$, then $K_r$ can be used as a key in a threshold operation.

## 1.1 Sharing Symmetric Operations

There are a few possible approaches to share a symmetric key primitive amongst a group of participants with a $(t, n)$-threshold access structure. We briefly review XOR-based approaches, examined in detail for block ciphers in [10] and for MACs in [6, 9]. An alternative is to use sequences or cascade ciphers (as in [4, 5]); for details on this approach, see [10]. Sharing using XOR has the advantage that the steps in the inverse operation need not be performed in the same order as the forward operation.

Our presentation is specialized to the $(t, n)$-threshold access structure, but these techniques are also applicable to general access structures. The methods are generic, in that they can be used with any secure block cipher or MAC, and the resulting shared primitives are at least as secure as the underlying function.

Let $\mathcal{K}$ be a keyspace, let $\mathcal{M}$ be a message space, let $\mathcal{T}$ be the set of authentication tags and let $F : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be a secure MAC. Suppose $K_r = (k_1, \ldots, k_t)$ is a set of keys held by some $A \in \Gamma$. The $t$-fold XOR MAC, $F^t : \mathcal{K}^t \times \mathcal{M} \to \mathcal{T}$ is defined as follows:

$$F^t(k_1, \ldots, k_t, m) = \left( \bigoplus_{i=1}^{t} F(k_i, m), r \right).$$

The index $r$ is also included for use during verification, in schemes with multiple keys $K_i$. To verify the tag $(\sigma, r)$ on the message $m$, the verifier computes $F(K_r, m) = (\sigma', r)$ and accepts the tag if $\sigma = \sigma'$. When referring to a *key of* $F^t$ we mean an element of $\mathcal{K}^t$, and refer to elements of $\mathcal{K}$ as *key components*. The following lemma proves that $F^t$ is at least as secure as $F$.

**Lemma 1.4** (Lemma 1, [10]). *If $F$ is a secure MAC, then $F^t$ is a secure MAC as well. Moreover, an adversary can generate a forged MAC for $F^t$ if and only if they know all key components $(k_1, \ldots, k_t)$.*

A similar construction is possible for block ciphers. Let $E : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ be a secure encryption function where $\mathcal{C}$ is the set of ciphertexts. A group of participants $A \in \Gamma$ can encrypt a message using $K_r = (k_1, \ldots, k_t)$ with

$$E^t(k_1, \ldots, k_t, m, n_0) = \left( m \bigoplus_{i=1}^{t} E(k_i, n_0), n_0, r \right),$$

where $n_0$ is a random nonce. Further details of these constructions and security proofs are given in [9, 10]. In this paper we will present schemes and give examples using a shared MAC; however, our results can also be applied to shared block ciphers.

## 1.2 The GCA-MAC Authentication Scheme

The following scheme is presented in [10]. We specialize our discussion to the case of a GCA$(t, n; \ell, t)$ constructed from a PHF$(\ell; n, t, t)$.

**Setup** Let $(Y, K_1, \ldots, K_t, \mathcal{B})$ be a GCA$(t, n; \ell, t)$, constructed from a PHF$(\ell; n, t, t)$. Let $F^t$ be the MAC defined in §1.1, and let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be the set of participants.

**Key Distribution** The receiver chooses a set of $\ell t$ key components and labels them with the elements of $Y$, then distributes the key components corresponding to $B_i$ to $P_i$. In other words, the receiver gives the key components indexed by column $i$ to $P_i$.

**Tag Creation** The participants $P_{i_1}, \ldots, P_{i_t}$ create a tag for a message $m$ as follows. First they determine $j$ such that $K_j \subseteq (B_{i_1} \cup \ldots \cup B_{i_t})$. This is done by finding a row in the $\ell \times t$ subarray of columns $i_1, \ldots, i_t$ which has $t$ distinct entries. If multiple rows have distinct entries, then $j$ is set to the first such row. Then the tag is computed as

$$(\sigma, j) = F^t(K_j, m),$$

and $m$ and $(\sigma, j)$ are sent to the receiver.

**Verification** The receiver uses $K_j$ to check if $(\sigma, j) \stackrel{?}{=} F^t(K_j, m)$ and accepts if they are equal.

## 1.3 GCA-MAC Example

We reproduce an example of a $(2, 8)$ GCA-MAC from [10], which will be used while discussing anonymity in Section 2. This GCA$(8, 2; 3, 2)$ is based on a

PHF$(3; 8, 2, 2)$.

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 |
| 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |

The three possible keys corresponding to the three rows are $K_1, K_2, K_3$, respectively, each constructed by participants having both a 1 and 2 in that row. This example is small enough to inspect all $\binom{8}{2} = 28$ possible cases. Note that pairs of participants can reconstruct one key (for example $P_1, P_2$), two keys (e.g. $P_2, P_3$) or all three keys (e.g. $P_1, P_8$). There are $\ell t = 6$ distinct key components.

# 2 Anonymity

Anonymity for shared symmetric key operations was first considered for the shared MAC schemes in [10]. There are two types of anonymity to consider. The first is *group anonymity*, which asks whether the receiver can learn which authorized subset collaborated to create the tag. The second type is *participant anonymity* (which we introduce in this paper), which asks whether the receiver can learn if a particular participant was involved in creating the tag. In both cases the receiver is given only the message and the tag output by the threshold scheme. In this section, we provide a general description of our new measures of anonymity, independent of any particular schemes, We use the GCA-MAC example given in Section 1.3 to illustrate the various concepts we discuss.

## 2.1 Threat Model

The goal is to prevent the adversary from learning which set of participants $A \in \Gamma$ performed a particular operation such as encryption or authentication. The adversary may be the receiver, who distributes key components and receives a ciphertext or authentication tag created by a group of participants. It may also be anyone knowing how keys were assigned to participants, who later observes the output of a shared primitive.

We *assume that the message does not leak the identity of the participants in* $A$ and that communication of the tag to the receiver is done using an anonymous channel. All details of communications between members of $\mathcal{P}$ are assumed to be hidden from the adversary. The knowledge of the adversary is limited to the output of the shared primitive, which includes all information required to perform the associated operation (decryption or verification). We also assume that all authorized sets are equally likely to use the primitive.

The anonymity provided is unconditional. Since multiple sets of participants can reconstruct the key used to perform the operation, they are all equally likely to have performed the operation in question, and no amount of computation on the part of the attacker will give additional information.

## 2.2 Group Anonymity

In this section we consider anonymity of a single group $A \in \Gamma$ and then define anonymity for a whole scheme $\mathcal{S}$.

### 2.2.1 Counting-Based Metrics

The counting-based metrics in this section extend the approach of Martin et al. [10]. The idea is to count the number of authorized sets which can reconstruct a key, and determine how likely each is to use it. Then, given that a particular key was used, determine which authorized sets were most likely to use it.

**Definition 2.1.** Let $\mathcal{S}$ be a $(t, n)$-threshold MAC over $n$ participants $\mathcal{P}$, with $\ell$ keys. The $(t, n)$ access structure is $\Gamma = \{A : A \subseteq \mathcal{P}, |A| = t\}$ and $|\Gamma| = \binom{n}{t}$. Let $A \in \Gamma$. For any message $m$ and valid tag $(\sigma, r)$ generated by $A$ we write $\Pr[A|r]$ to denote the conditional probability that $A$ created the given tag $(\sigma, r)$. The *degree of anonymity* of $A \in \Gamma$ with respect to $\mathcal{S}$ is defined $d(A, r) = 1 - \Pr[A|r]$.

The following definition of group anonymity is from [10], where it was originally called *anonymity*. We rename it *average degree of anonymity*.

**Definition 2.2.** In the notation of Definition 2.1, the *average degree of anonymity* for $\mathcal{S}$ is defined by

$$d_{av}(\mathcal{S}) = \frac{\sum \{d(A, r) : A \in \Gamma\}}{|\Gamma|} .$$

In the best case, upon seeing $m$ and $(\sigma, r)$, the adversary will see all sets in $\Gamma$ as being equally likely creators of $\sigma$. Therefore, at best, $d(A, r) = 1 - 1/\binom{n}{t}$. The average degree of anonymity of GCA-MAC was given in [10].

**Theorem 2.3.** *The GCA-MAC scheme has average degree of anonymity*

$$d_{av} \geq 1 - \frac{\ell}{\binom{n}{t}} .$$

Since a PHF$(\ell; n, t, t)$ can be constructed when $\ell \geq \lceil te^t \log n \rceil$ (see [12]), the following result holds.

**Theorem 2.4.** *There exists a $(t, n)$-threshold GCA-MAC scheme implemented with a* PHF$(\ell; n, t, t)$ *that has average degree of anonymity* $d_{av} \geq 1 - \lceil te^t \log n \rceil / \binom{n}{t}$.

As with security, anonymity should arguably be evaluated in the worst case, not the average case. We now present a new, stronger, counting-based anonymity metric.

**Definition 2.5.** Let $\mathcal{S}$ and $d(A)$ be as defined in Definition 2.1. The *anonymity* of $\mathcal{S}$ is defined

$$\mu = \min \{d(A, r) : A \in \Gamma, r \in [1, \ldots, \ell]\} .$$

Of course, average case anonymity is still a useful measure. If a scheme does not provide anonymity on average, anonymity in the worst case (and for all participants) will not be possible.

We now use the example of Section 1.3 to illustrate difference between $d_{av}$ and $\mu$. We also show how the GCA-MAC protocol reduces group anonymity by specifying that a $t$-set of participants $A$ will always choose the key corresponding to the *first* row separating $A$. Therefore, those sets of participants separated by rows one and two will never use key $K_2$. In the example above, keys $K_1, K_2$ and $K_3$ can each be reconstructed by 16 pairs of participants. However, since the GCA-MAC protocol selects the first separating row, we have the following:

- for $(\sigma, 1)$ there are 16 possible choices for $A$, hence $d(A, 1) = 1 - 1/16$,
- for $(\sigma, 2)$ there are 8 possible choices for $A$, hence $d(A, 2) = 1 - 1/8$, and
- for $(\sigma, 3)$ there are 4 possible choices for $A$, hence $d(A, 3) = 1 - 1/4$.

It can also be seen directly from the PHF that 16 inputs are separated first by row 1, eight are separated first by row 2 and four are separated first by row 3. Those $A \in \Gamma$ which use $K_3$ have an anonymity set one quarter the size of those which use $K_1$. While $d_{av} = 1 - \ell/\binom{8}{2} = 1 - 3/28 \approx 0.89$ (by Th. 2.3), the worst case anonymity is $\mu = 0.75$.

### 2.2.2 Entropy-Based Metrics

By computing the entropy of the conditional probability distributions of the previous section, we can measure the number of bits of uncertainty the adversary has about the creator of a given tag. This provides a more accurate measure.

**Definition 2.6.** Let $\mathcal{S}$ be as defined in Definition 2.1. Let $\mathbf{A}$ be a random variable defined on $\Gamma$, and let $r$ be the index of a key in $\mathcal{S}$. The *anonymity of key* $r$ is the entropy of the probability distribution $\Pr[\mathbf{A}|r]$:

$$h_r = H(\mathbf{A}|r) = -\sum_{A \in \Gamma} \Pr[A|r] \log_2 \Pr[A|r] \ .$$

If the adversary observes a tag created with key $r$, then $h_r$ can be interpreted as the number of bits of uncertainty the adversary has about the group which created the tag. It measures the effect on group anonymity caused by using different keys. In our GCA-MAC example, $h_1 = 4$, $h_2 = 3$ and $h_3 = 2$, which is consistent with the adversary's knowledge (e.g. given that $K_3$ was used, he can be certain that it was created by one of four groups). More generally, $\mu \approx 1 - 2^{-\min\{h_r : r \in \mathcal{S}\}}$ since this represents the worst case for anonymity. Note that an upper bound on $h_r$ is the log of the number of groups separated by row $r$, in this case four bits. The sets separated by row 1 are provided best possible anonymity under this metric. In general, if $T_r$ groups are separated by row $r$, then $T_r$ is size of the largest anonymity set possible given that $r$ was used. Therefore, key entropy close to $\log_2(T_r)$ bits is desirable.

Averaging over the possible keys that $A$ may use gives the average anonymity provided to $A$.

**Definition 2.7.** Let $\mathcal{S}$ be as defined in Definition 2.1. The *average anonymity provided to* $A \in \Gamma$ is

$$\mu_{av}(A) = \sum_{r=1}^{\ell} (\Pr[r|A] \times h_r) \ . \tag{1}$$

The *average anonymity of the scheme* $\mathcal{S}$ is defined to be the average of $\mu_{av}(A)$ for all groups $A \in \Gamma$,

$$\mu_{av}(\mathcal{S}) = \frac{1}{|\Gamma|} \sum_{A \in \Gamma} \mu_{av}(A) \ .$$

Since in GCA-MAC groups only use one key, the average anonymity of $A$ coincides with the key anonymity for the first row which separates $A$. The average anonymity of the scheme is $\frac{1}{28}(16 \cdot 4 + 8 \cdot 3 + 4 \cdot 2) = 3.4$ bits. There is also a natural relation, as we saw between $h_r$ and $\mu$, to the average degree of anonymity, namely $d_{av}(\mathcal{S}) \approx 1 - 2^{\mu_{av}(\mathcal{S})}$.

**Remark 2.8.** If all rows have the same value for $h_r$, then $\mu_{av}(A) = h_r$, since $\sum_{r=1}^{N} \Pr[A|r] = 1$. In this case the average anonymity is the same for all $A \in \Gamma$. This is desirable so that no "weak keys" exist with respect to anonymity, i.e. keys with extremely low entropy (such as $K_3$ in our GCA-MAC example). When $h_r$ is the same for all keys, we have $\mu_{av}(\mathcal{S}) = \mu_{av}(A) = h_r$, for all $A \in \Gamma$. The improved scheme we will present in Section 3 has this property.

## 2.3 Participant Anonymity

The issue of participant anonymity has not been considered in previous work. Participant anonymity is more challenging to provide than group anonymity, since there are only $n$ participants, while there are $\binom{n}{t}$ groups.

Let $m$ be a message and $(\sigma, r)$ be a valid tag created by an unknown group $A \in \Gamma$, observed by the receiver. For $P_i \in \mathcal{P}$, let $\Pr[P_i|r]$ be the probability that $P_i \in A$ (the group that created the tag), given that $r$ was used. Suppose $\mathcal{S}$ has $\ell$ keys. We define the *participant anonymity of* $P_i \in \mathcal{P}$ to be

$$\rho(P_i) = 1 - \max\{\Pr[P_i|r] : r \in [1, \ldots, \ell]\} \ ,$$

and the *participant anonymity of the scheme* $\mathcal{S}$ to be

$$\rho(\mathcal{S}) = \min\{\rho(P_i) : P_i \in \mathcal{P}\} \ .$$

With respect to $\rho$, it is desirable for $\mathcal{S}$ to have two properties. First, $\rho(\mathcal{S}) = 1 - t/n$ is best possible, since if all participants are equally likely, and $t$ are required for an operation, then a participant has probability $t/n$ of being involved. Therefore, it is desirable that $\rho(\mathcal{S})$ be close to $1 - t/n$. Also, participant anonymity should be *equitable*, that is $\rho(P_i)$ should not differ significantly from $\rho(P_j)$ (for all $P_i, P_j \in \mathcal{P}$). Unfortunately, in any scheme constructed from a PHF, since $\ell = \Omega(\log n)$, there is a trade-off between efficiency and participant anonymity.

## 2.4 Malicious Setup Attack on Anonymity

In this attack, the scheme is set up so that anonymity is reduced for certain participants. The attack works by adding a row to the PHF that separates a small number of $A \in \Gamma$. Suppose we use the following PHF$(4; 9, 3, 3)$ to create a GCA$(3, 9; 4, 3)$ (source: PHFtables [13]).

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 2 | 2 | 3 | 2 | 3 | 1 | 1 |
| 1 | 3 | 1 | 3 | 1 | 2 | 2 | 2 | 3 |
| 1 | 2 | 2 | 1 | 3 | 3 | 1 | 2 | 3 |
| 3 | 3 | 2 | 1 | 1 | 3 | 2 | 1 | 2 |

(2)

Using the GCA-MAC protocol,

- given $(\sigma, 1)$, there are 27 possible choices for $A$,
- given $(\sigma, 2)$, there are 21 possible choices for $A$,
- given $(\sigma, 3)$, there are 18 possible choices for $A$,
- given $(\sigma, 4)$, there are 18 possible choices for $A$.

If the attacker adds a "dummy" row to the PHF,

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 |
| 1 | 3 | 2 | 2 | 3 | 2 | 3 | 1 | 1 |
| 1 | 3 | 1 | 3 | 1 | 2 | 2 | 2 | 3 |
| 1 | 2 | 2 | 1 | 3 | 3 | 1 | 2 | 3 |
| 3 | 3 | 2 | 1 | 1 | 3 | 2 | 1 | 2 |

,

then any message authenticated using $K_1$ must have $A = \{P_8, P_9, P_i\}$ (where $P_i$ is any of the other participants). This reduces the number of possible groups to $7, 27, 18, 16, 16$ for keys $1, 2, 3, 4, 5$, respectively. Using the measures defined above, $\mu = 1 - 1/18 = 0.94$ before the attack and $\mu = 1 - 1/7 = 0.86$ after the attack. The effect on participant anonymity is that $\rho(P_8) = \rho(P_9) = 0$ since the receiver can say for sure that $P_8$ and $P_9$ participated whenever $K_1$ is used. The key associated with the "dummy row" has much lower key entropy than the others, reducing uncertainty about which groups use this key.

This attack may be especially effective when combined with other information about the participants. If the attacker knows which participants are most likely to initiate a message, they will be frequent senders, and the attack will be more effective when they are targeted during setup.

## 2.5 Verifiable Setup

More generally, the setup should be verified by the participants to ensure consistency of the key components distributed by the dealer. For example, in (2),

suppose the key component given to $P_1$, call it $k'_{1,1}$, in the first row differs from $k_{1,1}$ given to $P_8$ and $P_9$ (who also have a 1 in row 1). Since $P_1$ is the only participant who holds $k'_{1,1}$, upon receiving $(\sigma, 1)$ the receiver may check whether $k'_{1,1}$ was used to create $(\sigma, 1)$ and learn whether $P_1$ belongs to the set which created it.

A simple approach to verifying consistency is to have each pair of participants engage in a key confirmation protocol to ensure they hold the same keys (when they should, based on the PHF). If a trusted bulletin board is available, the dealer might alternately publish a cryptographic hash of the key component and the associated PHF to allow participants to check the validity of their key components. In this work we simply assume the setup has been verified for consistency, and leave the problem of efficient verification to future work.

# 3 An Improved Scheme: BPHF-MAC

The new scheme given below, BPHF-MAC, makes two changes to GCA-MAC to improve anonymity. First, when multiple rows separate $A \in \Gamma$, i.e. when $A$ can reconstruct multiple keys $K_i$, a row/key is chosen at random from all of those possible. This recovers the anonymity lost when only the first row which separates $A$ is used.

The second change is to counter the malicious setup attack, and prevent keys which provide weak anonymity. A special type of hash family, introduced by Stinson [15], will be used in BPHF-MAC instead of an arbitrary PHF.

**Definition 3.1.** A PHF$(\ell; n, m, t)$ is *balanced* if in every row, each symbol occurs exactly $n/m$ times. The notation BPHF$(\ell; n, m, t)$ is used to denote a balanced PHF.

The balance property will maximize the possible key entropy, and simplify our analysis of group and participant anonymity of BPHF-MAC. Fortunately, many good explicit constructions of PHFs are balanced. A PHF is said to be *linear* if the code formed by taking the columns as codewords is linear. Since any linear PHF is balanced, constructions from Reed-Solomon codes [14] or AG-codes [17] give BPHF. The examples given in Section 1.3 and 2.4 are balanced.

**BPHF-MAC:**

**Setup** Construct a BPHF$(\ell; n, t, t)$, where $f_i : X \to Y$. Let $F^t$ be the MAC defined in §1.1, and let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be the set of participants. The authorized sets are $\Gamma = \{A \in \mathcal{P} : |A| = t\}$. The BPHF should be verified by the participants if it is constructed by a party untrusted with respect to anonymity.

**Key Distribution** The receiver chooses a set of $\ell t$ key components and labels them with the elements of $Y$, creates $\mathcal{B}$ using the construction of §1, and distributes the key components corresponding to $B_i$ to $P_i$. In other words, the receiver gives key components indexed by column $i$ to $P_i$.

**Tag Creation** The participants $P_{i_1}, \ldots, P_{i_t}$ create a tag for a message $m$ as follows. First they determine the set $J$ such that $K_j \subseteq (B_{i_1} \cup \ldots \cup B_{i_t})$ for all $j \in J$. This is done by finding the rows in the $\ell \times t$ subarray of columns $i_1, \ldots, i_t$ which have distinct entries. Since we are using a PHF of strength $t$ we are guaranteed $|J| \geq 1$. An index $j$ is chosen uniformly at random from $J$, and the tag is computed to be

$$(\sigma, j) = F^t(K_j, m) = F^t(k_1, \ldots, k_t, m) \ ,$$

and $(\sigma, j)$ is sent to the receiver as the tag for $m$.

**Verification** The receiver uses $K_j$ to check if $(\sigma, j) \stackrel{?}{=} F^t(K_j, m)$ and accepts if they are equal.

We remark that the performance of the tag creation step may be improved by repeatedly selecting a row at random and checking whether it separates the participants, until a separating row is found. There is no need for the set $J$ to be computed explicitly.

The security of **BPHF-MAC** follows from the security of $F^t$ (Lemma 1.4) and the properties of a GCA.

## 3.1 Anonymity of BPHF-MAC

To evaluate the anonymity of **BPHF-MAC** we first prove a key property of BPHF.

**Lemma 3.2.** *In a* BPHF$(\ell; n, m, t)$*, with functions* $f_i : X \to Y$*, the size of the set* $\{A \subset X : |A| = t, |f_i(A)| = t\}$ *is*

$$\binom{m}{t}\left(\frac{n}{m}\right)^t \ ,$$

*for all* $f_i$*,* $1 \leq i \leq \ell$*.*

*Proof.* From the definition of a BPHF, each of the $m$ symbols occurs $n/m$ times in each row. We ask how many $t$-sets of columns are distinct when restricted to row $i$. The $t$ symbols can be chosen in $\binom{m}{t}$ ways, and for each of these $t$ symbols we must choose one of the $n/m$ positions. Therefore in a BPHF$(\ell; n, m, t)$ each row separates $\binom{m}{t}(n/m)^t$ sets of $t$ columns. □

The following definition will also be used to analyze the anonymity of BPHF-MAC.

**Definition 3.3.** Let $\mathcal{C}$ be a code, and $A$ be a $t$-set of codewords of $\mathcal{C}$. The *$t$-separating distance*, denoted $s_{A,t}$, is the number of coordinates in which all $t$ codewords in $A$ differ.

The $t$-separating distance first appears in the work of Bassalygo et al. [2] (where it was originally called the $t$-th hash distance). The $t$-separating distance may be seen as a generalization of the classic Hamming distance, which is the same as the 2-separating distance. A PHF$(\ell; n, m, t)$ guarantees $s_{A,t} \geq 1$ while a $\lambda$-PHF$(\ell; n, m, t)$ guarantees $s_{A,t} \geq \lambda$ ([7, 16]). In the $(t, n)$ BPHF-MAC scheme, $t$ is the threshold size, so we will simply write $s_A$ in what follows.

To compute the anonymity of the BPHF-MAC scheme we require knowledge of $\Pr[A|r]$, the probability that $A$ created a tag using key $r$ (row $r$ of the PHF). Let us first consider $\Pr[r|A]$. If $A$ is not separated by row $r$, it cannot use key $r$, therefore

$$\Pr[r|A] = \begin{cases} 0 & \text{when } r \text{ does not separate } A \\ \frac{1}{s_A} & \text{when } r \text{ does separate } A\,. \end{cases}$$

The probability that row $r$ is used is $1/s_A$ since $A$ will choose $r$ at random from one of the $s_A$ separating rows.

We have assumed that $\Pr[A] = 1/\binom{n}{t}$, i.e., all sets of participants are equally likely to create a tag. The probability $\Pr[r]$, i.e. the probability that row $r$ is used for a tag, is given by

$$\begin{aligned} \Pr[r] &= \sum_{A \in \Gamma} (\Pr[r|A] \times \Pr[A]) \\ &= \frac{1}{\binom{n}{t}} \sum_{\substack{A \in \Gamma \\ r \text{ separates } A}} \frac{1}{s_A} \end{aligned} \tag{3}$$

Since $s_A$ is at most $\ell$ and the sum in (3) has $n^t/t^t$ terms by Lemma 3.2,

$$\begin{aligned} \Pr[r] &\geq \frac{1}{\binom{n}{t}} \left(\frac{n^t}{t^t}\right) \frac{1}{\ell} \\ &= \frac{n^t}{\ell t^t \binom{n}{t}}\,. \end{aligned}$$

From Bayes' theorem,

$$\begin{aligned} \Pr[A|r] &= \frac{\Pr[r|A]\Pr[A]}{\Pr[r]} \\ &= \frac{(1/s_A)\left(1/\binom{n}{t}\right)}{\Pr[r]} \\ &= \frac{1}{s_A \binom{n}{t} \Pr[r]\,.} \end{aligned} \tag{4}$$

Now recall that $\mu = 1 - \max\{\Pr[A|r] : A \in \Gamma, r = 1, \ldots, \ell\}$. The probability $\Pr[A|r]$ is maximized when the denominator of (4) is smallest, i.e. when $s_A = 1$

and $\Pr[r] = \frac{n^t}{\ell t^t \binom{n}{t}}$. Therefore

$$\begin{aligned}\mu &= 1 - \frac{1}{\binom{n}{t}\frac{n^t}{\ell t^t \binom{n}{t}}} \\ &= 1 - \frac{\ell t^t \binom{n}{t}}{\binom{n}{t} n^t} \\ &= 1 - \frac{\ell t^t}{n^t} \ . \end{aligned} \tag{5}$$

Recall that for GCA-MAC, $d_{av} = 1 - \ell/\binom{n}{t}$. Since $(\ell t!)/(n^t) \leq \ell/\binom{n}{t} \leq (\ell t^t)/n^t$, the worst-case anonymity of BPHF-MAC is comparable to the average case anonymity of GCA-MAC. If $t$ is fixed, they are asymptotically equal.

**Example 3.4.** As an example, we compute $\mu$ for the $(2, 8)$ BPHF-MAC scheme implemented with the PHF$(3; 8, 2, 2)$ given in Section 1.3, which has $\ell = 3$, $n = 2$, $t = 2$:

$$\mu = 1 - \frac{3(2^2)}{8^2} = 1 - \frac{12}{64} = 0.8125 \ .$$

The average anonymity of GCA-MAC, which we computed in Section 2.2.1, was 0.89.

### 3.1.1 Key Anonymity and Cyclic BPHF

We now consider the key anonymity of BPHF-MAC. n the following, we will require knowledge of $s_A$ values for each $A$ separated by $r$. This motivates the following definition. Let $\vec{S}_r$ be the length $\ell$ vector defined as

$$\vec{S}_r(i) = |\{A \in \Gamma : A \text{ is separated by row } r \text{ and } s_A = i\}| \ .$$

$\vec{S}_r$ is the *distribution of separating distances* for $t$-sets separated by row $r$. Ignoring the values in $\vec{S}_r$, we prove that a large class of BPHF have $\vec{S}_{r_i} = \vec{S}_{r_j}$ for all rows $r_i, r_j$. This implies $h_{r_i} = h_{r_j}$, which is a desirable property for anonymity (recall Remark 2.8). The class in question are BPHF constructed from cyclic codes, which we briefly review here.

**Definition 3.5.** Let $\mathcal{C}$ be a code of length $\ell$ with symbols from an alphabet $\Sigma$, and let $(c_1, c_2, \ldots, c_{\ell-1}, c_\ell) \in \Sigma^\ell$. $\mathcal{C}$ is *cyclic* if

$$c = (c_1, c_2, \ldots, c_{\ell-1}, c_\ell) \in \mathcal{C}$$

implies

$$c' = (c_\ell, c_1, \ldots, c_{\ell-2}, c_{\ell-1}) \in \mathcal{C} \ .$$

The transformation of a codeword from $c$ to $c'$ is called a *cyclic shift*. If a PHF is constructed from a cyclic code, we say it is a *cyclic* PHF.

Important classes of cyclic codes are BCH codes (which include Reed-Solomon codes) and quadratic residue codes. Reed-Solomon codes with large distance are an easily constructed class of cyclic BPHF (see Stinson et al. [14]). The example of Section 1.3 is a cyclic BPHF.

**Theorem 3.6.** *In a cyclic* BPHF$(\ell; n, m, t)$, $\vec{S}_i = \vec{S}_j$ *for all* $i, j \in \{1, \ldots \ell\}$.

*Proof.* The case $i = j$ is trivial. Without loss of generality, choose a pair $(i, j)$ where $j > i$, $j = i + g$. Define the following two sets of $t$-sets of columns separated by rows $i$, and $j$,

$$X_i = \{A_1, \ldots, A_T\}$$
$$X_j = \{A'_1, \ldots, A'_T\}$$

where $T = (n/t)^t$. The balance property of the PHF (see Lemma 3.2) provides the value of $T$ and proves $T$ is the same for all rows.

Consider $\phi : X_i \to X_j$ and define $\phi(A)$ as the set of columns obtained by cyclically shifting each column in $A$ by $g$ positions. This mapping is well defined, i.e. $\phi(A)$ is a set of columns in the BPHF by the cyclic property, and $\phi(A)$ is separated by row $j$, since $j = i + g$ and $A$ is separated by row $i$. We now show that

(i) $\phi$ preserves $s_A$ values, i.e., $s_A = s_{\phi(A)}$, and

(ii) $\phi$ is one-to-one.

Property (i) holds since the rows are shifted but not modified, so a row separating (or not separating) $A$ is intact with a different index in $\phi(A)$. Since the separating distance is the same, $s_A = s_{\phi(A)}$. It is clear that $\phi(A_n) \neq \phi(A_m)$ for all $n \neq m$ since $A_n \neq A_m$. Therefore the image of $\phi$ in $X_j$ has size $T$, which implies $\phi$ is one-to-one.

Since the $t$-sets of columns separated by row $i$ are in one-to-one correspondence with those separated by row $j$, and they have the same $s_A$ values, $\vec{S}_i = \vec{S}_j$. □

The following theorem is the implication of Theorem 3.6 on anonymity in BPHF-MAC.

**Theorem 3.7.** *Let* $\mathcal{S}$ *be an instance of the BPHF-MAC scheme constructed with a cyclic BPHF. Then* $\mu_{av}(A_i) = \mu_{av}(A_j) = \mu_{av}(\mathcal{S})$ *for all* $A_i, A_j \in \Gamma$.

*Proof.* In the case of cyclic codes we can show that all rows are equally likely to be used in a tag. The probability that a given row $r$ is used (recall equation (3)) is:

$$\begin{aligned}\Pr[r] &= \frac{1}{\binom{n}{t}} \sum_{\substack{A \in \Gamma \\ r \text{ separates } A}} \frac{1}{s_A} \\ &= \frac{1}{\binom{n}{t}} \left( \frac{\vec{S}_r(1)}{1} + \frac{\vec{S}_r(2)}{2} + \ldots + \frac{\vec{S}_r(\ell)}{\ell} \right) .\end{aligned}$$

This quantity is the same for all rows since $\vec{S}_{r_i} = \vec{S}_{r_j}$ for all $r_i, r_j$ and hence $\Pr[r_i] = \Pr[r_j]$. Substituting $\Pr[r] = 1/\ell$ in (4) gives

$$\Pr[A|r] = \frac{\ell}{s_A \binom{n}{t}} ,$$

and $h_r$ can be expressed as follows:

$$h_r = - \sum_{\substack{A \in \Gamma \\ r \text{ separates } A}} \frac{\ell}{s_A \binom{n}{t}} \log_2 \left( \frac{\ell}{s_A \binom{n}{t}} \right) .$$

We will group the terms of this sum by $s_A$ value. There are $\ell$ possible $s_A$ values, and $\vec{S}_r(i)$ is the number of terms (i.e. sets $A$) with $s_A = i$. Therefore,

$$h_r = - \sum_{i=1}^{\ell} \vec{S}_r(i) \frac{\ell}{i \binom{n}{t}} \log_2 \left( \frac{\ell}{i \binom{n}{t}} \right) . \tag{6}$$

Since the values $\vec{S}_r$ are are the same for all rows, $h_r$ is also the same for all rows. This is sufficient, by Remark 2.8, to show that the entropy-based measures of group anonymity are equal. □

While we are not able to compute $\mu_{av}$ and $h_r$ for BPHF-MAC when arbitrary BPHF are used, we can assure that, for cyclic BPHF, anonymity will be equitable. Computing $\vec{S}_r$ for large codes/PHF appears to be a difficult problem.

**Example 3.8.** For the cyclic BPHF$(3; 8, 2, 2)$ given as an example in Section 1.3, $\vec{S}_1 = \vec{S}_2 = \vec{S}_3 = (4, 8, 4)$ since there are 4, 8 and 4 sets separated by exactly 1, 2, and 3 rows, respectively. Since this BPHF is cyclic, and $\vec{S}$ is known, we can use (6) to compute the key entropy of any row $r$:

$$\begin{aligned} h_r &= - \sum_{i=1}^{3} \vec{S}_r(i) \frac{3}{i \binom{8}{2}} \log_2 \left( \frac{3}{i \binom{8}{2}} \right) \\ &= -4 \left( \frac{3}{28} \log \frac{3}{28} \right) + 8 \left( \frac{3}{56} \log \frac{3}{56} \right) + 4 \left( \frac{3}{84} \log \frac{3}{84} \right) \\ &= 3.9 . \end{aligned}$$

As noted in the discussion following Definition 2.6, four bits is the maximum possible key entropy in this example, therefore BPHF-MAC provides nearly optimal anonymity for all $A \in \Gamma$.

### 3.1.2 Bounding the Key Entropy

While computing $h_r$ for BPHF-MAC schemes constructed from arbitrary BPHF appears difficult without knowledge of $\vec{S}_r$, we can use the min-entropy of $\Pr[A|r]$ to guarantee a minimum amount of anonymity.

Let $X$ be a random variable defined on a set $\mathcal{X}$ and $\gamma = \max_{x \in \mathcal{X}} \{\Pr[X = x]\}$. The *min-entropy* of $X$ is defined to be

$$H_\infty(X) = \log_2 \left(\frac{1}{\gamma}\right) = -\log_2 \gamma \, .$$

Since $H(X) \geq H_\infty(X)$, the min-entropy gives a lower bound on the Shannon entropy.

With respect to the key anonymity of BPHF-MAC, defined as $H(A|r)$, to compute $H_\infty(A|r)$, we must determine the maximum value of $\Pr[A|r]$ over all authorized sets $A$ and all rows $r$. Following equation (5) in Section 3.1, we have

$$\max \{P[A|r] : A \in \Gamma, 1 \leq r \leq \ell\} = \frac{\ell t^t}{n^t}$$

and therefore

$$\begin{aligned} h_r &\geq \log_2 \left(\frac{n^t}{\ell t^t}\right) \\ &= t \log_2 n - \log_2 l - t \log_2 t \, . \end{aligned} \tag{7}$$

While this lower bound will, in general, be strictly lower than the actual value of $h_r$, it is useful since it can be easily computed for any instance of BPHF-MAC, without knowledge of $\vec{S}_r$.

**Example 3.9.** Recall the example PHF$(3; 8, 2, 2)$ given in Section 1.3, for which $h_r$ was computed in Example 3.8 as 3.9 bits. Using the bound of (7), we can compute

$$\begin{aligned} h_r &\geq 3 \log_2 8 - \log_2 2 - 2 \log_2 2 \\ &= 2.4 \end{aligned}$$

A similar computation gives $h_r \geq 2.75$ for the PHF$(4; 9, 3, 3)$ given in Section 2.4, while $h_r = 4.6$. In Example 3.13, for $C_{2,2}$, $h_r \geq 1$ while $h_r = 1.91$, for $C_{2,6}$, $h_r \geq 7.41$ while $h_r = 9.87$, and for $C_{2,10}$, $h_r \geq 14.68$ while $h_r = 17.92$.

### 3.1.3 A Code with Known Separating Distance Distribution

Here we describe a simple code for which $\vec{S}_r$ may be computed explicitly. Let $C_{q,\ell}$ be the $(\ell; q^\ell, q)$ *complete code* over $q$ symbols of length $\ell$. Note that $C_{q,\ell}$ is cyclic since the cyclic shift of any codeword is another $\ell$-tuple of the $q$ symbols, hence it belongs to $C_{q,\ell}$.

**Theorem 3.10.** *In the code $C_{q,\ell}$ for a coordinate $r$, the separating distance distribution vector $\vec{S}_r$ defined*

$$\vec{S}_r(i) = |\{A : A \subset C_{q,\ell}, |A| = q, A \text{ is separated by } r \text{ and } s_{A,q} = i\}|,$$

*has the values*

$$\vec{S}_r(i) = \binom{\ell - 1}{i - 1} (q!)^{i-1} (q^q - q!)^{\ell - i + 1} \, ,$$

*for $i = 1, \ldots, \ell$.*

*Proof.* We wish to count the number of $q$-sets of codewords separated by coordinate $r$ with separating distance $i$. The coordinate $r$ is fixed, and contains symbols $1, \ldots, q$. Each $q$-set will have $i-1$ separating coordinates (excluding $r$) and $\ell - i + 1$ non-separating coordinates. First we must choose which $i-1$ coordinates will be separated, which can be done in $\binom{\ell-1}{i-1}$ distinct ways. Each of the $i-1$ separating coordinates may be chosen in $q!$ ways, since they must be a permutation of $\{1, \ldots, q\}$. The $\ell - i + 1$ non-separating coordinates can be chosen in $q^q - q!$ different ways, since they are all possible assignments minus those which separate the codewords in this coordinate. Taking the product gives the desired result; the number of $q$-sets of codewords in $C_{q,\ell}$ which are separated in position $r$ and have $i-1$ other separating coordinates. □

The following corollary applies Theorem 3.10 to the case $q = 2$.

**Corollary 3.11.** *$C_{2,\ell}$ is a cyclic* BPHF$(\ell, 2^\ell, 2, 2)$ *with*

$$\vec{S}_r(i) = \binom{\ell-1}{i-1} 2^{\ell-1}$$

*for all rows $r$ and $i = 1, \ldots \ell$.*

*Proof.* Note that $C_{2,\ell}$ is a PHF of strength 2 because all codewords are distinct, and that $C_{q,\ell}$ is cyclic as remarked above. $C_{2,\ell}$ is balanced since it contains all words of length $\ell$ over the alphabet $\{0, 1\}$. □

The next theorem gives an explicit formula for our entropy-based anonymity measures for the case when $C_{2,\ell}$ is used with the BPHF-MAC scheme, by applying the formulae of Theorem 3.7.

**Theorem 3.12.** *Let $\mathcal{S}$ be an instance of the $(2, 2^\ell)$ BPHF-MAC scheme implemented with $C_{2,\ell}$. Then*

$$\mu_{av}(A) = \mu_{av}(\mathcal{S}) = h_r = -\sum_{i=1}^{\ell} \binom{\ell-1}{i-1} 2^{\ell-1} \frac{\ell}{i\binom{n}{t}} \log_2 \left( \frac{\ell}{i\binom{n}{t}} \right)$$

*for all $A \in \Gamma$ and key indices $r$.*

*Proof.* Equality of $\mu_{av}(A), \mu_{av}(S)$ and $h_r$ follows from Theorem 3.7. Recall that in the case of cyclic BPHF, following (6), we can express $h_r$ as:

$$h_r = -\sum_{i=1}^{\ell} \vec{S}_r(i) \frac{\ell}{i\binom{n}{t}} \log_2 \left( \frac{\ell}{i\binom{n}{t}} \right) . \tag{8}$$

By substituting $\vec{S}_r(i)$ with the value given in Corollary 3.11, we arrive at the desired formula. □

**Example 3.13.** We give a few examples of the group and participant anonymity ($\mu$ and $\rho$) as well as the average anonymity $\mu_{av}$, provided by the $(2, 2^\ell)$ BPHF-MAC scheme implemented with $C_{2,\ell}$. Details of the participant anonymity of BPHF-MAC are given in Section 3.2. The column $\mu_{av}(opt)$ gives $\log_2((n/t)^t)$ which is the largest possible value of $h_r$ (see the discussion following Definition 2.6).

| $\ell$ | $t$ | $n$ | $\mu$ | $\mu_{av}(\mathcal{S})$ | $\mu_{av}(opt)$ | $\rho$ |
|---|---|---|---|---|---|---|
| 2 | 2 | 4 | 0.5 | 1.91 bits | 2 bits | 0.5 |
| 6 | 2 | 64 | 0.994 | 9.87 bits | 10 bits | 0.968 |
| 10 | 2 | 1024 | 0.99996 | 17.92 bits | 18 bits | 0.998 |

## 3.2 Participant anonymity of BPHF-MAC

In this section we determine the anonymity of individual participants. Analysis of the participant anonymity provided by BPHF-MAC is simpler than group anonymity, and relies only on the balance property. BPHF-MACs provide optimal and equitable participant anonymity, as proven in the following theorem.

**Theorem 3.14.** *The participant anonymity of BPHF-MAC is*

$$\rho(P_j) = 1 - \frac{t}{n}$$

*for all $P_j \in \mathcal{P}$.*

*Proof.* First, recall that every row separates $(n/t)^t$ sets of participants (Lemma 3.2). Let $P_j$ be any participant, and $r$ be any row. Given that $K_r$ was used, we evaluate $\Pr[P_j|r]$, the probability that $P_j$ has participated in the creation of a tag using $K_r$. $P_j$ has some symbol in row $r$, hence there remain $t-1$ symbols corresponding to participants which can belong to a set including $P_j$ separated by row $r$. Since our PHF is balanced, each of these symbols occurs $(n/t)$ times in row $r$. The other $t-1$ symbols/participants can be chosen in $(n/t)^{t-1}$ ways. Therefore,

$$\begin{aligned}
\Pr[P_j|r] &= \frac{|\{A \in \Gamma : P_j \in A,\ \text{row } r \text{ separates } A\}|}{|\{A \in \Gamma : \text{row } r \text{ separates } A\}|} \\
&= \frac{(n/t)^{t-1}}{(n/t)^t} \\
&= \frac{1}{n/t} \\
&= \frac{t}{n}
\end{aligned}$$

Since $\Pr[P_j|r] = t/n$ is the same for all rows and all participants, $\rho(P_j) = 1 - t/n$ for all $P_j \in \mathcal{P}$ as required. □

# 4 GCA Constructions From Arbitrary PHF

When previously discussing constructions of GCAs using PHF, we gave only the connection between PHF$(\ell; n, t, t)$ and GCA$(t, n; \ell, \ell)$. This is a restriction, since we require that the number of symbols in the PHF and the strength be equal (both must be $t$).

It is also possible to construct GCAs from a general perfect hash family, denoted PHF$(\ell; n, m, t)$, where $m \geq t$. The new structure is only a GCA by some definitions, due to the following small difference. Some definitions[1] require that $\mathcal{K} = \{K_1, \ldots, K_v\}$ (the set of keys) be a *partition* of $Y = \{k_1, \ldots, k_{\ell m}\}$ (the set of key components). In the construction we are about to describe this is not the case, however all other GCA properties are satisfied. A GCA where the sets of $\mathcal{K}$ are not necessarily disjoint, which we call a *relaxed* GCA, will be shown adequate for the application at hand.

We use the following PHF$(3; 12, 5, 3)$ to illustrate the construction (source: PHFtables [13]).

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_7$ | $P_8$ | $P_9$ | $P_{10}$ | $P_{11}$ | $P_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 4 | 4 | 0 | 1 | 0 | 2 | 3 | 3 | 3 | 1 |
| 3 | 1 | 0 | 4 | 0 | 1 | 2 | 0 | 2 | 4 | 3 | 3 |
| 1 | 4 | 0 | 1 | 3 | 2 | 1 | 2 | 4 | 2 | 3 | 0 |

As in the restricted case, each (row, symbol) pair will correspond to a key component. The total number of key components is $\ell m$ in general, and 15 in our example. Instead of having only one key $K_r$ associated to row $r$, there will be $\binom{m}{t}$ keys; one key for each subset of $t$ symbols in row $r$. The total number of keys is therefore $\ell\binom{m}{t}$. In the example, each row has a key associated with each of the ten 3-sets of key components:

$$\{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,5\},$$
$$\{1,3,5\}, \{2,3,5\}, \{1,4,5\}, \{2,4,5\}, \{3,4,5\}$$

for a total of $3 \times 10 = 30$ keys. There are $n$ players, and player $i$ is given the key components corresponding to the $i$-th column, denoted $B_i$. The notation $K_{r,i}$ refers to the $i$-th key of row $r$. The subsets are numbered using $m$ bits in the following canonical way: if symbol $j$ $(1 \leq j \leq m)$ belongs to the subset then bit $j$ is 1, otherwise bit $j$ is zero. For example, $\{1,2,3\}$ is numbered $(00111)_2 = 7$ and $\{2,4,5\} = (11010)_2 = 26$.

Now we verify the GCA properties.

1. For a set $A = \{P_{i_1}, \ldots, P_{i_t}\} \in \Gamma$, columns $i_1, \ldots, i_t$ will be distinct in at least one row, therefore the key components held by $A$: $\bigcup_{j=1}^{t} B_{i_j}$, will contain at least one $K_{r,i}$. $A$ may have one key from multiple rows, but never multiple keys from any row, since in each row they hold only $t$ key components.

[1] The definition of GCAs in the literature is inconsistent. In [9] and [8] it is not specified that the $K_i$s be disjoint, but in [11] and [10] the requirement that the $K_i$s be disjoint appears. The fact that the $K_i$s are disjoint is used once in an anonymity proof in [10].

2. For a set $A = \{P_{i_1}, \ldots, P_{i_{t-1}}\} \notin \Gamma$, none of the $K_{r,i}$ are held since $A$ holds at most $t-1$ key components from each row, and $|K_{r,i}| = t$ (for all keys).

Note that the necessity of the condition that the keys in $\mathcal{K}$ be disjoint depends crucially on how the key components are distributed to participants. The relaxed GCA construction ensures that no participant gets more than one component of any $K_{r,i}$, to preserve the threshold condition.

For an example, take participants $P_5, P_7$ and $P_8$. They have key components 0 and 2 in row 1, components 0 and 2 in row 2, and components 1, 2 and 3 in row 3. Therefore, they may use key $K_{3,7}$. Further, this key depends on components held by all three participants and hence this key could not be used by only one or two of the participants.

The only modification required to the BPHF-MAC scheme when using a relaxed GCA is that the index of the key used must be computed differently. The index information in the tag grows from $\log(\ell)$ bits to $\log(\ell)+m$ bits (using the subset numbering scheme given above, but this can be reduced to $t\log(m)$ or fewer bits if desired).

Using a hash family with these parameters for the BPHF-MAC scheme will improve efficiency significantly, as shown in Section 4.1, but at the cost of a small reduction in anonymity, discussed in Section 4.2.

**A note on strong unforgeability.** This MAC scheme does not provide *strong unforgeability*, as defined by An, Dodis and Rabin [1]. While it is not possible for an adversary to create a valid tag for a new message, it may be possible to create a different tag on a previously authenticated message. Consider a $(2, n)$ scheme which has four key components per row, fix a row, and denote the key components $k_1, \ldots, k_4$. Given $\sigma_1 = F_{k_1}(m) \oplus F_{k_2}(m)$ and $\sigma_2 = F_{k_3}(m) \oplus F_{k_2}(m)$, the tag $\sigma_1 \oplus \sigma_2$ is valid but different for the same message $m$.

## 4.1 Impact on Efficiency

We make three general remarks with respect to efficiency of BPHF-MAC schemes based on relaxed GCAs.

1. When considering upper bounds for PHF [3], larger $n$ are possible when $\ell$ and $t$ are fixed, since $n \leq t^{\ell/(t-1)} < m^{\ell/(t-1)}$ when $m > t$.

2. There are a better variety of constructions available, including those from coding theory, which construct cyclic BPHF.

3. The potentially large number of keys $\ell\binom{m}{t}$ has no effect on efficiency, since the *number of key components* is only $\ell m$.

Intuitively, lifting the requirement that all keys $K_i$ be disjoint (as tuples of key components), increases the number keys available for a given number of key components, which reduces the number of such components required.

The comparison relevant for BPHF-MAC is the number key components for a $(t, n)$ scheme using a PHF$(\ell; n, t, t)$, versus the number of key components

when a PHF$(\ell'; n, m, t)$ is used. Asymptotically, the improvement will be at most a constant factor and will depend on a specific construction, since for fixed $m$ and $t$, $\ell = O(\log n)$.

**Example 4.1.** We compare the number of key components required for a $(5, 121)$ threshold scheme. Using the PHF construction based on Reed-Solomon codes in [14], we can construct a PHF$(11; 121, 11, 5)$, and BPHF-MAC requires $11 \times 11 = 121$ key components in total, while each participant must store 11 key components. The best construction of a PHF$(\ell, 121, 5, 5)$ on PHFtables [13] has $\ell = 176$. Here, BPHF-MAC requires $176 \times 5 = 880$ key components in total, about 7.3 times more. Each participant must store 176 key components, which is 16 times more than the Reed-Solomon construction.

**Example 4.2.** We repeat example 4.1 but using $t = 6$. Using the Reed-Solomon construction we obtain a PHF$(16; 256, 16, 6)$, which is trivially also a PHF$(16; 121, 16, 6)$. The $(6, 121)$ and $(6, 256)$ BPHF-MAC schemes using this PHF require 256 key components in total, while each participant must store 16 key components. The equivalent best known construction of PHF$(\ell; 121, 6, 6)$ from [13] has $\ell = 1160$ for a total of 6960 total key components and a storage requirement of 1160 key components per participant. In the $(6, 256)$ case, the lowest value of $\ell$ is 1232, giving a total of 7392 key components, and a storage requirement of 1232 key components per participant.

Additional examples with larger $t$ for comparison are difficult to construct due to the lack of direct constructions for PHF$(N; n, t, t)$. If we consider the existence result of Mehlhorn [12], which states that a PHF$(\ell; n, m, t)$ exists when $\ell \geq te^{t^2/m} \log n$, we can make a more general comparison. In the case that $m = \alpha t$, this bound requires $\ell \geq t \sqrt[\alpha]{e^t} \log n$, so the minimum value of $\ell$ is reduced from $te^t \log n$ to $t \sqrt[\alpha]{e^t} \log n$.

## 4.2 Impact on Anonymity

To determine the worst-case anonymity of BPHF-MAC based on relaxed GCAs (the $m \geq t$ case), we use an approach similar to the one used when $m = t$ in Section 3.1. Let $r_i$ denote the $i$-th key of row $r$ (recall that there are $\binom{m}{t}$ keys per row). By "$r_i$ separates $A$" we mean that (i) row $r$ separates $A$, and (ii) $A$ has the $i$-th $t$-set of symbols in row $r$. In other words, $A$ is not only separated by $r$, but separated by $t$ specific symbols in row $r$.

Given a group $A \in \Gamma$, $A$ may use $s_A$ keys, and will choose to use one of them with probability $1/s_A$. Therefore,

$$\Pr[r_i|A] = \begin{cases} 0 & \text{when } r_i \text{ does not separate } A \\ \frac{1}{s_A} & \text{when } r_i \text{ separates } A\,. \end{cases}$$

Now we consider $\Pr[r_i]$ the probability that key $r_i$ is used.

$$\begin{aligned}\Pr[r_i] &= \sum_{A\in\Gamma} (\Pr[r_i|A] \times \Pr[A]) \\ &= \frac{1}{\binom{n}{t}} \sum_{\substack{A\in\Gamma \\ r_i \text{ separates } A}} \frac{1}{s_A} \end{aligned} \tag{9}$$

The number of $A \in \Gamma$ separated by a given $r_i$ is $(n/m)^t$ in a BPHF, since each of the $t$ symbols in $r_i$ appears $n/m$ times in row $r$. Since $s_A$ is at most $\ell$ and the sum in (9) has $n^t/m^t$ terms,

$$\begin{aligned}\Pr[r_i] &\geq \frac{1}{\binom{n}{t}} \left(\frac{n^t}{m^t}\right) \frac{1}{\ell} \\ &= \frac{n^t}{\ell m^t \binom{n}{t}} \, .\end{aligned}$$

From Bayes' theorem,

$$\begin{aligned}\Pr[A|r_i] &= \frac{\Pr[r_i|A]\Pr[A]}{\Pr[r_i]} \\ &= \frac{(1/s_A)\left(1/\binom{n}{t}\right)}{\Pr[r_i]} \\ &= \frac{1}{s_A \binom{n}{t} \Pr[r_i]} \end{aligned} \tag{10}$$

Now recall that $\mu = 1 - \max\{\Pr[A|r_i] : A \in \Gamma, r_i = 1, \ldots, \ell\binom{m}{t}\}$. The probability $\Pr[A|r_i]$ is maximized when the denominator of (10) is smallest, i.e. when $s_A = 1$ and $\Pr[r_i] = \frac{n^t}{\ell m^t \binom{n}{t}}$. Therefore

$$\begin{aligned}\mu &= 1 - \frac{1}{\binom{n}{t} \frac{n^t}{\ell m^t \binom{n}{t}}} \\ &= 1 - \frac{\ell m^t}{n^t} \, ,\end{aligned}$$

which corresponds to equation 5 in the case $m = t$.

For fixed $\ell$, there is clearly a decrease in anonymity. However, as shown in the efficiency discussion, setting $m > t$ reduces $\ell$. Therefore, when $m > t$, $\ell$ decreases while the other term in the numerator increases. We now present two examples, one where anonymity is significantly decreased, and one where it is decreased only slightly.

**Example 4.3.** Recall the example PHF$(3; 8, 2, 2)$ from Section 1.3. Using this PHF, the $(2, 8)$ BPHF-MAC scheme has $\mu = 0.81$. We can replace this PHF with the following PHF$(2; 8, 4, 2)$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 1 | 2 | 2 | 3 | 3 | 4 | 4 |

This instance of the $(2, 8)$ BPHF-MAC scheme has $\mu = 0.5$.

**Example 4.4.** Using the same parameters as in Example 4.1, the $m = t$ instance of the $(5, 121)$ BPHF-MAC has $\mu = 0.9999787$ and $h_r \geq 15.53$, while the relaxed instance has $\mu = 0.9999316$ and $h_r \geq 13.84$. (The bounds on $h_r$ are given by the min-entropy of $\Pr[A|r_i]$.)

When we revisit the $(6, 256)$ scheme from 4.1, we find $\mu = 0.99999979$ and $h_r \geq 22.22$ if $m = t$. The $(6, 256)$ scheme based on the relaxed construction has $\mu = 0.99999904$ and $h_r \geq 20$.

The decrease in anonymity can be explained by the relative sizes of the parameters in the examples. In Example 4.3, $|\Gamma|$ is much smaller than in Example 4.4. Also the impact of the change in $\ell$ in Example 4.4 counterbalances some of the lost anonymity.

**Participant Anonymity.** An analysis similar to the proof of Theorem 3.14 shows that $(t, n)$ BPHF-MAC schemes constructed with $\mathrm{PHF}(N; n, m, t)$ have participant anonymity $\rho(P) = 1-m/n$ for all participants $P$. This is a reduction from $1 - t/n$.

# 5 Conclusion

We have strengthened the definition of anonymity in the context of shared symmetric key primitives. Group anonymity is measured in the worst case, and the concept of participant anonymity was introduced. We have presented modified schemes for sharing symmetric key operations with improved group and participant anonymity using balanced perfect hash families. The relaxed GCA construction of Section 4 provides a useful trade-off for practical applications, providing large gains in efficiency with only a small decrease in anonymity.

# References


[1] J.H. An, Y. Dodis and T. Rabin. On the security of joint signature and encryption. *Proceedings of EUROCRYPT '02, LNCS* **2332** (2002), 83–107.

[2] L.A. Bassalygo, M. Burmester, A. Dyachkov and G. Kabatianski. Hash codes. *Proceedings of the 1997 IEEE International Symposium on Information Theory* (1997), 174.

[3] S.R. Blackburn and P.R. Wild. Optimal linear perfect hash families. *Journal of Combinatorial Theory, Series A* **83** (1998), 233–250.

[4] E.F. Brickell, G. Di Crescenzo and Y. Frankel. Sharing block ciphers. *Information Security and Privacy, LNCS* **1841** (2000), 457–470.

[5] S. Even and O. Goldreich. On the power of cascade ciphers. *ACM Transactions on Computer Systems* **3** (1985), 108–116.

[6] J. Katz and A.Y. Lindell. Aggregate message authentication codes. *Proceedings of CT-RSA '08, LNCS* **4964** (2008), 155–169.

[7] L. Liu and H. Shen. Explicit constructions of separating hash families from algebraic curves over finite fields. *Designs, Codes and Cryptography*, **41** (2006), 221–233.

[8] S. Long, J. Pieprzyk, H. Wang and D.S. Wong. Generalised cumulative arrays in secret sharing. *Designs, Codes and Cryptography* **40** (2006), 191–209.

[9] K.M. Martin, R. Safavi-Naini, H. Wang and P.R. Wild. Distributing the encryption and decryption of a block cipher. *Designs, Codes and Cryptography* **36** (2005), 263–287.

[10] K.M. Martin, J. Pieprzyk, R. Safavi-Naini, H. Wang and P.R. Wild. Threshold MACs. *Proceedings of ICISC 2002*, *LNCS* **2587** (2003), 237–252.

[11] K.M. Martin and S.-L. Ng. The combinatorics of generalised cumulative arrays. *Journal of Mathematical Cryptology* **1** (2007), 13–32.

[12] K. Melhorn. *Data Structures and Algorithms*, Vol. 1, Springer-Verlag (1984).

[13] R.A. Walker II. PHFtables.com. `www.phftables.com`. Accessed April 2008.

[14] D.R. Stinson, R. Wei and L. Zhu. New constructions for perfect hash families and related structures using combinatorial designs and codes. *Journal of Combinatorial Designs* **8** (2000), 189–200.

[15] D.R. Stinson. Some baby-step giant-step algorithms for the low hamming weight discrete logarithm problem. *Mathematics of Computation* **71** (2002), 379–391.

[16] D.R. Stinson and R. Wei. Generalized cover-free families. *Discrete Mathematics* **279** (2004), 463–477.

[17] H. Wang and C. Xing. Explicit constructions of perfect hash families from algebraic curves over finite fields. *Journal of Combinatorial Theory, Series A* **93** (2001), 112–124.

[18] H. Wang and J. Pieprzyk. Shared generation of pseudo-random function with cumulative maps. *Proceedings of CT-RSA '03*, *LNCS* **2612** (2003), 281–294.