Multi-receiver Homomorphic Authentication Codes for Network Coding

Zhaohui Tang, and Hoon Wei Lim

Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
TANG0209@e.ntu.edu.sg, hoonwei@ntu.edu.sg

Abstract. We investigate a new class of authenticate codes (A-codes) that support verification by a group of message recipients in the network coding setting. That is, a sender generates an A-code over a message such that any intermediate node or recipient can check the authenticity of the message, typically to detect pollution attacks. We call such an A-code as *multi-receiver homomorphic A-code* (MRHA-code). In this paper, we first formally define an MRHA-code. We then derive some lower bounds on the security parameters and key sizes associated with our MRHA-codes. Moreover, we give efficient constructions of MRHA-code schemes that can be used to mitigate pollution attacks on network codes. Unlike prior works on computationally secure homomorphic signatures and MACs for network coding, our MRHA-codes achieve unconditional security.

Keywords: Authentication codes, network coding, homomorphism, unconditional security.

1 Introduction

Network Coding In recent years, network coding [6] has received considerable attention for its ability to improve a network's throughput and robustness. This is achieved by enabling an intermediate node within a network to encode its incoming messages before forwarding them, as opposed to more straightforward traditional store-and-forward routing technology. To illustrate this, let us consider a very simple scenario where a sender (or source) wishes to transmit a file V to a group of recipients. To do this through the classic *linear network coding*,[1] the sender first cuts the file into m messages $\mathbf{v}_1, \ldots, \mathbf{v}_m$ of length n, where $n > m$. The messages are represented as linearly independent vectors with coefficients in a finite field $\mathbb{F}_q$, that is, $\mathbf{v}_i \in \mathbb{F}_q^n$ for $1 \leq i \leq m$. When a linear network code over $\mathbb{F}_q$ is used, every intermediate node in the network computes an $\mathbb{F}_q$-linear combination of the received messages from its incoming edges, before forwarding the resulting linear combination to its outgoing edge(s). For each vector $\mathbf{v}_i$, the sender then appends a vector $\mathbf{e}_i$, a whole zero vector of length m with a '1' at the i-th position. (This is required for decoding at the recipients.) The vectors $(\mathbf{v}_i, \mathbf{e}_i) \in \mathbb{F}_q^{n+m}$ for $1 \leq i \leq m$ are then sent over the network, and each recipient uses the last m symbols to recover the associated message vectors from the linear transformation. For a successful recovery of the original file V, clearly a recipient needs to obtain m non-corrupted linearly independent vectors encoding V.

Pollution Attacks One consequence of letting intermediate nodes process messages before forwarding them is that, the messages are now inevitably susceptible to data modification or

[1] We consider only linear network coding in this paper.

corruption. This has a severe effect, as the corrupted messages can be propagated to other downstream nodes in the network preventing the recipients from recovering the correct messages, and thus, is known as a *pollution attack*. To see this, we slightly abuse the notation by using V to denote an m-dimensional subspace $V = \text{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$. An adversary can corrupt one or more messages by creating either a new vector $\mathbf{v} \notin V$, or an entire m-dimensional subspace $V' \neq V$, or even modifying $\mathbf{v} \in V$ such that $\mathbf{v}' \notin V$. All the resulting corrupted messages can be combined into legitimate messages making their way to the destination and finally cause decoding failures at the recipients. Message authentication is, therefore, of particular interest and importance in network coding.

However, it is not at all clear how classical authentication schemes can be used to protect message authenticity in network coding. This is so since messages are mixed in transit, the received messages are most likely different from those sent, and thus one cannot verify a message authentication tag (or signature) without accessing the original messages. One way to get around this is through a homomorphic authentication scheme, as recently proposed by Boneh *et al.* [3]. The homomorphic property allows any intermediate node to perform a linear combination of tags corresponding with some outgoing messages without decoding the incoming messages. Moreover, verification can be performed directly on mixed messages by the intermediate node or the recipient. In this paper, we focus on message authentication in the network coding setting, aiming to detect a pollution attack on encoded messages in transit through a network.

Our Approach We consider an information-theoretic secure message authentication scheme. We assume that each node in the network shares a private key with the source. The shared key can be independent from or has some dependency with the others. Our scheme then generates unconditionally secure authentication codes that allow any intermediate node and the recipient to verify their received messages. Our work starts from authentication codes (A-codes) [9] in the information-theoretic setting (see Section 2.2 for more details). Particularly, we extend the concept of multi-receiver A-codes (MRA-codes) proposed by Safavi-Naini and Wang [11] with the homomorphic property, hence the moniker *multi-receiver homomorphic A-codes* (MRHA-codes).

Roughly speaking and informally, for N verifiers (inclusive of all intermediate nodes and recipients), an MRHA-code consists of $N + 1$ homomorphic A-codes (associated with the sender and the N verifiers). Moreover, for each verifier R_i for $1 \leq i \leq N$, the A-code associated with R_i can be derived from the A-code associated with the sender (through some mapping functions). Our scheme then defines: (i) a tag generation algorithm for the sender and the intermediate node, (ii) a verification algorithm for the verifier R_i, and (iii) for each R_i, a unique mapping function from the sender's key space to the verifier's key space. The tag generation algorithm is defined such that any intermediate node is able to generate a tag corresponding to a linear combination of incoming messages (without decoding the incoming messages). On the other hand, the verification algorithm is defined such that the tag corresponding to a message $\mathbf{v}$ verifies successfully if and only if $\mathbf{v} \in V$, where V is the linear subspace originated at the sender. This requires the verifier's private key (shared with the sender) and mapping function. (We formally define an MRHA-code and our scheme in Section 3.)

With regards to security, we consider the case of one or more untrusted verifiers, *i.e.* insider attacks. These adversaries may exploit their own key information to send rogue messages on behalf of the source. We consider an attack to be succeeded if any (honest) verifier accepts one of the rogue messages as authentic. Our goal is to design a scheme that is not only

unconditionally secure against such an attack, but also *efficient* in terms of key sizes and the number of tags transmitted between two nodes.

We note that there exists a simple, but very inefficient, solution to our goal. If we assume that all the private keys for the scheme are *independent* from each other, we can trivially use the A-code from [13] that supports *single-receiver* verification: the source shares an independent key with each verifier and appends one tag per verifier per message. If there are N verifiers, this trivial scheme prevents any collusion among $N-1$ verifiers from cheating the N-th verifier.[2] However this requires the source to store a large number of keys and the intermediate node to transmit a large number of tags. In our solution, we permit *some dependencies* among the N keys, while ensuring the required security, assuming that any collusion among $\mu - 1$ verifiers, where $\mu < N$, is possible. This significantly reduces the key size of the source and the number of tags to be transmitted over the network.

Related Work Prior to our work, Oggier and Fathi [10] also investigated mitigation of pollution attacks in network coding by means of information-theoretic secure A-codes. As with the setting we consider, they constructed A-codes that allow intermediate nodes and recipients of a network to verify the authenticity of messages. However, the key difference is that they did not consider homomorphic A-codes. As we define in Section 2.2, the tag generation algorithm for a homomorphic A-code must be a linear mapping from the source (message) space to the tag space. Consequently, rather than employing simple linear transformations in homomorphic A-codes, the scheme of [10] involves a huge number of exponential operations (nm^2 of exponentiations in $\mathbb{F}_q$) for tag generation and thus is likely to be too expensive to implement in practice. Yet another difference between Oggier and Fathi's scheme and ours is that, the former is required to take as input the network coding coefficients—an additional parameter compared to ours. (Hence, each message is required to carry one additional bit to keep track of the network coefficients.)

Independently, Tang [13] recently defined a homomorphic A-code in the context of network coding. However, the homomorphic A-code considered in [13] can be used in only a very basic setting, that is each A-code is designed to be verifiable by only one target recipient in the network. As explained, extending this straightforwardly to the multi-receiver setting would result a very inefficient scheme.

There have been more works on *computationally secure* homomorphic schemes in the cryptographic literature. In the asymmetric key setting, homomorphic signature schemes have been proposed, see for example [3, 5, 7]. These signature schemes typically work on linear subspaces. A signature is considered to be valid if and only if it is an element of a predefined subspace, while it is difficult to forge a signature that is not an element of the defined subspace. One key advantage of homomorphic signature schemes is that they have simpler key management. All verifiers in the network are required to possess only the public key of the source. Nevertheless, for some applications, asymmetric key operations may be considered computationally expensive. This motivates proposals of alternative solutions in the symmetric key setting—homomorphic MAC schemes [1, 8, 2]. While these schemes are more efficient than signature schemes, they require distribution of private keys among all the verifiers in the network, a considerable bigger effort than distributing just the sender's public key.

[2] Note that in the trivial scheme, the colluded $N-1$ verifiers cannot do better than an *outsider* who does not share any private key with the source.

CONTRIBUTIONS We first give a formal definition of a multi-receiver homomorphic A-code (MRHA-code). Given some suitable security parameters, we then derive some useful bounds: (i) information-theoretic bounds for a general MRHA-code (see Theorem 1), and (ii) lower bounds on the key sizes at the sender and the receivers in some security parameters under some constraint (see Theorem 2). We present the definition and the proofs of the bounds in Section 3. Further, using an homomorphic A-code as a building block, we construct a class of MRHA-codes which are significantly more efficient than the non-homomorphic scheme in [10] and the aforementioned trivial scheme derived from [13] (see Theorem 3). Our A-codes use very simple and efficient linear mappings for tag generation; and our construction requires $\mu < N$ keys at the sender and μ tags for each message, in comparison with N keys at the sender and N tags per message required by the trivial scheme. We particularly show an MRHA-code meeting the lower key size bounds (see Theorem 4 and Corollary 1). We present the details of these results in Section 4.

2 Preliminaries

We recall some basic notions of A-codes [9, 15] and homomorphic A-codes [13].

2.1 Notation

For the remainder of the paper, we use similar notation from [11] and [14]. For any random variable X, Y, Z, we let $P(x)$ denote the probability distribution when $X = x$; let $P(x, y)$ denote the probability distribution when $X = x$ and $Y = y$; and let $P(y|x)$ denote the conditional probability of $Y = y$ when provided $X = x$. Further, we let $H(X)$ represent the entropy of X; let $H(Y|X)$ be the conditional entropy of Y given X; let $I(Y; X)$ be the mutual information between Y and X; and let $I(Z; Y|X)$ be the conditional mutual information of Z and Y given X.

2.2 Authentication codes (A-codes)

A *systematic* A-code (or A-code *without secrecy*) [9] consists of a quadruple $(\mathcal{S}, \mathcal{K}, \mathcal{A}, f)$ where $\mathcal{S}, \mathcal{K}, \mathcal{A}$ denote the source (message) space, key space and tag space, respectively, while f is a function defined to be: $\mathcal{S} \times \mathcal{K} \rightarrow \mathcal{A}$.

Briefly, an A-code is used for message authentication as follows: the sender and the receiver secretly share a common private (or secret) key $\mathbf{k} \in \mathcal{K}$. To send a message $\mathbf{v} \in \mathcal{S}$, the sender first generates a tag $\mathbf{t} = f(\mathbf{v}, \mathbf{k}) \in \mathcal{A}$ over the message and transmits the message-tag pair $(\mathbf{v}, \mathbf{t})$ to the receiver. The receiver then checks the authenticity of the received message $\mathbf{v}$ by verifying whether or not the message-tag pair $(\mathbf{v}, \mathbf{t})$ satisfies $\mathbf{t} = f(\mathbf{v}, \mathbf{k})$. If the equality holds, $\mathbf{v}$ is accepted; otherwise $\mathbf{v}$ is rejected.

In terms of security, we typically consider two types of attacks: (i) *impersonation attack*, where an adversary attempts to insert a new message-tag pair without prior observation, and (ii) *substitution attack*, where an adversary first observes a valid message-tag pair $(\mathbf{v}, \mathbf{t})$ and then attempts to insert $(\mathbf{v}', \mathbf{t}')$ with $\mathbf{v} \neq \mathbf{v}'$.

Wang *et al.* [15] proposed A-codes that are linear in keys and showed that they are useful in distributed authentication schemes. The key linearity allows an authentication key to be shared among a group of verifiers. A-codes that are linear in messages, however, are usually avoided in classic A-codes. This is because the linearity in messages opens up the possibility

for an adversary to forge a message with a valid tag, by simply computing a linear combination of observed message-tag pairs. Nevertheless, this limitation turns out be useful for message authentication in the context of network coding, as described in the following subsection.

2.3 Homomorphic A-Codes

In [13], Tang introduced a (q, n, m)-homomorphic A-code, which is an A-code linear in messages, to authenticate an m-dimensional subspace $V \subseteq \mathcal{S}$ where $\mathcal{S}$ is an n-dimensional vector space over $\mathbb{F}_q$.

Definition 1. *(Definition 1 of [13]) An A-code $(\mathcal{S}, \mathcal{K}, \mathcal{A}, f)$ is a (q, n, m)-homomorphic A-code if*

i) $\mathcal{S}$ and $\mathcal{A}$ are finite-dimensional vector spaces over $\mathbb{F}_q$, with $\mathsf{dim}(\mathcal{S}) = n$,
ii) for every m-dimensional subspace $V \subseteq \mathcal{S}$, and every $\mathbf{v} = \sum_{i=1}^{m} \alpha_i \mathbf{v}_i \in V$ where $\alpha_i \in \mathbb{F}_q (1 \leq i \leq m)$, $f(\mathbf{v}, \mathbf{k})$ satisfies

$$f(\sum_{i=1}^{m} \alpha_i \mathbf{v}_i, \mathbf{k}) = \sum_{i=1}^{m} \alpha_i f(\mathbf{v}_i, \mathbf{k}).$$

The second property is simply a rephrase of the fact that $(\mathcal{S}, \mathcal{K}, \mathcal{A}, f)$ is linear in messages, where we assume that $\mathcal{S} = \mathbb{F}_q^n$, $\mathcal{A} = \mathbb{F}_q^t$ and $\mathcal{K} \subseteq \mathbb{F}_q^{n \times t}$. From Definition 1, it is not difficult to see that the mapping f in a (q, n, m)-homomorphic A-code $(\mathcal{S}, \mathcal{K}, \mathcal{A}, f)$ must be an $\mathbb{F}_q$-linear mapping from $\mathcal{S}$ to $\mathcal{A}$. In other words, we can always write $f(\mathbf{v}, \mathbf{k}) = \mathbf{v}M_{\mathbf{k}}$ with $M_{\mathbf{k}}$ an $n \times t$ matrix determined by $\mathbf{k}$. (In a general A-code, the mapping f is not necessarily a linear mapping from source space to tag space [10].)

Recall that the homomorphic A-code of [13] is designed for verification by only a target recipient. Given a homomorphic A-code $(\mathcal{S}, \mathcal{K}, \mathcal{A}, f)$ and assume that the recipient shares a private key $\mathbf{k} \in \mathcal{K}$ with the source, message authentication is then carried out as shown in Figure 1.

- **Tag Generation:** The sender generates a tag $\mathbf{t}_i = f(\mathbf{v}_i, \mathbf{k}) \in \mathcal{A}$ for each of $\mathbf{v}_i$ for $i = 1, \ldots, m$.
- **Combination:** Assume that each intermediate node receives some message-tag pair $(\mathbf{x}_j, \mathbf{t}_{\mathbf{x}_j})$ for some index j, where each $\mathbf{x}_j$ is already a linear combination of $\mathbf{v}_i$. The intermediate node computes $\sum_j \alpha_j \mathbf{t}_{\mathbf{x}_j}$ as the tag $\mathbf{t}_{\mathbf{y}}$ corresponding to an output vector $\mathbf{y} = \sum_j \alpha_j \mathbf{x}_j$, where the sum is over some subset of the received tags.
- **Verification:** The recipient takes as input a received message-tag pair $(\mathbf{v}, \mathbf{t})$ and the shared key $\mathbf{k}$, and checks if $\mathbf{t} = f(\mathbf{v}, \mathbf{k})$. If the equation holds, the recipient accepts $\mathbf{v}$; otherwise it rejects. (Note that indeed, any output $\mathbf{v}$ can be written as $\sum_{i=1}^{m} \alpha_i \mathbf{v}_i$.)

Fig. 1. Definition of homomorphic A-code scheme.

Here, the impersonation attack described in Section 2.2 can be further refined into a *message* and a *subspace impersonation* attacks. The former is, as before, to attempt to forge a valid tag for a message $\mathbf{v} \neq \mathbf{0} \in \mathcal{S}$; while the latter is to forge a valid tag for a previously unseen m-dimensional subspace $V = \mathrm{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$. Moreover, we consider a third type of attack called *subspace substitution* attack, where, as implied by its name, the adversary attempts to forge a valid tag for message $\mathbf{v}$ when he observes a tag for an m-dimensional subspace $V = \mathrm{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$ with $\mathbf{v} \notin V$.

See [13] for further details on the security analysis of a homomorphic A-code against these attacks, and the trade-off between efficiency and security.

3 Multi-receiver Homomorphic A-codes

We now turn to main crux of this paper.

3.1 Definitions

We assume there are N verifiers in total in the network wanting to authenticate their incoming messages. These, denoted by $R_1, \ldots, R_N$, include all the intermediate nodes and the recipients at the destinations. We now formally define an MRHA-code.

Definition 2. *Let $\mathcal{C} = (\mathcal{S}, \mathcal{A}, \mathcal{K}, f)$ and $\mathcal{C}_i = (\mathcal{S}, \mathcal{A}_i, \mathcal{K}_i, f_i)$ be (q, m, n)-homomorphic A-codes for $1 \leq i \leq N$. We call $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ a (q, m, n, N)-MRHA-code if for each $1 \leq i \leq N$, there exists an $\mathbb{F}_q$-linear mapping $\pi_i : \mathcal{A} \to \mathcal{A}_i$ and another mapping $\tau_i : \mathcal{K} \to \mathcal{K}_i$ such that for any $(\mathbf{v}, \mathbf{k}) \in \mathcal{S} \times \mathcal{K}$, we have:*

$$f_i(\mathbf{v}, \tau_i(\mathbf{k})) = \pi_i(f(\mathbf{v}, \mathbf{k})). \tag{1}$$

We assume that for each homomorphic A-code $\mathcal{C}_i$, the probability distribution on the source space of $\mathcal{C}_i$ is the same as that in the code $\mathcal{C}$, and the probability distribution on $\mathcal{K}_i$ is derived from that of $\mathcal{K}$ and the mapping τ_i. We can now see that, using the properties of homomorphism in $\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N$ and linearity in π_i, an (q, m, n, N)-MRHA-code can indeed be used by any verifier R_i to verify the authenticity of a message from the sender, provided that condition (1) holds.

Building on the above definition, we then specify our MRHA-code scheme comprising four algorithms as shown in Figure 2 (recall that a file is split into m pieces $\mathbf{v}_i \in \mathcal{S}$ for $1 \leq i \leq m$ before it is transmitted over the network).

- **Key distribution:** A trusted authority (or the source itself) randomly chooses a private key $\mathbf{k} \in \mathcal{K}$ for the source. For each verifier R_i (for $1 \leq i \leq N$), the trusted authority sends $\tau_i(\mathbf{k})$ as R_i's private key.
- **Tag generation:** For each message $\mathbf{v}_i \in \mathcal{S}$ $(1 \leq i \leq m)$, the source computes $f(\mathbf{v}_i, \mathbf{k})$ as the corresponding tag $\mathbf{t}_i$ and sends out the message-tag pair $(\mathbf{v}_i, \mathbf{t}_i)$.
- **Combination:** Assume that each intermediate node receives some message-tag pair $(\mathbf{x}_h, \mathbf{t}_{\mathbf{x}_h})$ for some index h, where each $\mathbf{x}_h$ is already a linear combination of some messages $\mathbf{v}_i$. The intermediate node computes $\sum_h \alpha_h \mathbf{t}_{\mathbf{x}_h}$ as the tag $\mathbf{t}_{\mathbf{y}}$ corresponding to an output vector $\mathbf{y} = \sum_h \alpha_h \mathbf{x}_h$, where the sum is taken over some subset of the received tags.
- **Verification:** Assume that a verifier R_j possesses a private key $\tau_i(\mathbf{k})$ and it receives a message $\mathbf{v}$ and the corresponding tag $\mathbf{t}_{\mathbf{v}}$. The verifier checks if $f_i(\mathbf{v}, \tau_i(\mathbf{k})) = \pi_i(\mathbf{t}_{\mathbf{v}})$; it accepts $\mathbf{v}$ as authentic if the equation holds; otherwise it rejects.

Fig. 2. Definition of MRHA-code scheme.

The *correctness* of the verification algorithm follows immediately from Definition 2. Since $\mathcal{C}$ is homomorphic, the intermediate node can generate tags for their outgoing messages by simply combing their incoming tags. Further, since $\mathcal{C}, \mathcal{C}_i$ are homomorphic, π_i is $\mathbb{F}_q$-linear and τ_i, π_i satisfy condition (1), verification by R_i can be performed by checking if $f_i(\mathbf{v}, \tau_i(\mathbf{k})) = \pi_i(\mathbf{t}_{\mathbf{v}})$ holds.

It is also worth noting that when $\mathcal{C}_1 = \mathcal{C}$ in Definition 2, we obtain a single-verifier homomorphic A-code scheme, as defined in Section 2.3.

3.2 Security

We require some further notation to analyze our MRHA-codes. Given an (q, m, n, N)-MRHA-code $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$, we represent a random variable $\widetilde{\Upsilon}$ as the collection of all message-tag pairs in $\mathcal{C}$, that is

$$\widetilde{\Upsilon} = \{\upsilon = (\mathbf{v}, \mathbf{t}) : \mathbf{v} \in \mathcal{S}, \mathbf{t} \in \mathcal{A}\}.$$

For any $(\mathbf{v}, \mathbf{t}) = \upsilon \in \widetilde{\Upsilon}$, we say υ is valid for R_i if there exists a key $\mathbf{k} \in \mathcal{K}$ such that $f_i(\mathbf{v}, \tau_i(\mathbf{k})) = \pi_i(\mathbf{t})$. We then represent Υ as the collection of all the message-tag pairs from $\mathcal{C}$ and valid for R_i. Furthermore, we introduce $\Upsilon^m = \Upsilon \times \cdots \times \Upsilon$ to denote the collection of all m-tuple elements, each of which is from Υ. We say a sequence of the form

$$\overline{\upsilon} = (\overline{\upsilon_1}, \cdots \overline{\upsilon_m}) \in \Upsilon^m \text{ where } \overline{\upsilon_j} = (\overline{\mathbf{v}_j}, \overline{\mathbf{t}_j}) \text{ for } 1 \leq j \leq m$$

is valid for R_i if $\overline{\upsilon_i}$'s are linearly independent from each other in $\mathbb{F}_q$ and there exists a key $\mathbf{k} \in \mathcal{K}$ such that

$$f_i(\overline{\mathbf{v}_j}, \tau_i(\mathbf{k})) = \pi_i(\overline{\mathbf{t}_j}) \text{ for } 1 \leq j \leq m.$$

We next use $\overline{\Upsilon}^m = \overline{\Upsilon_1} \times \cdots \times \overline{\Upsilon_m}$ to denote the collection of all such valid sequences $\overline{\upsilon} = (\overline{\upsilon_1}, \cdots, \overline{\upsilon_m})$. For any sequence $(\overline{\upsilon_1}, \cdots, \overline{\upsilon_m}) = \overline{\upsilon} \in \overline{\Upsilon}^m$, we introduce another element $\upsilon' \in \Upsilon$, where $\upsilon' = (\mathbf{v}', \mathbf{t}')$. We say that υ' is valid for R_i when $\overline{\upsilon}$ is observed by R_L under the condition that $\mathbf{v}' \notin \text{span}(\overline{\mathbf{v}_1}, \cdots, \overline{\mathbf{v}_m})$ and there exists a key $\mathbf{k} \in \mathcal{K}$ such that

$$f_i(\overline{\mathbf{v}_j}, \tau_i(\mathbf{k})) = \pi_i(\overline{\mathbf{t}_j}) \text{ and } f_i(\mathbf{v}', \tau_i(\mathbf{k})) = \pi_i(\mathbf{t}') \text{ for } 1 \leq j \leq m.$$

Finally, we use Υ' to denote the collection of all such υ''s for $\overline{\Upsilon}^m$.

We now consider the security of our MRHA-codes. We let $R_L = \{R_{i_1}, \ldots, R_{i_l}\}$ to denote a group of corrupted or malicious verifiers, where $L = \{i_1, \ldots, i_l\} \subseteq \{1, \ldots, N\}$. We let also $\mathcal{K}_L$ to denote $\mathcal{K}_{i_1} \times \cdots \times \mathcal{K}_{i_l}$ and τ_L to denote $\tau_{i_1} \times \cdots \times \tau_{i_l}$. Our security goal is to prevent the following three types of attacks from R_L on R_i (where $i \notin L$):

- *Message impersonation attack*: Given the relevant private keys, but without any prior observation, the goal of adversary R_L is to create a message $\mathbf{v} \in \mathcal{S}$ such that it is accepted as authentic by verifier R_i. The success probability of R_L in the message impersonation attack is expressed as:

$$P_I[i, L] = \max_{\mathbf{k}_L \in \mathcal{K}_L} \max_{\upsilon \in \widetilde{\Upsilon}} P(\upsilon \text{ is valid for } R_i | \mathbf{k}_L). \tag{2}$$

- *Subspace impersonation attack*: This is similar with a message impersonation attack, except that here, the goal of adversary R_L is to create an m-dimensional subspace $V = \text{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$ that is accepted as authentic by verifier R_i. The success probability of R_L in the subspace impersonation attack is defined to be:

$$P_{I_{sub}}[i, L] = \max_{\mathbf{k}_L \in \mathcal{K}_L} \max_{\overline{\upsilon} \in \Upsilon^m} P(\overline{\upsilon} \text{ is valid for } R_i | \mathbf{k}_L). \tag{3}$$

- *Subspace substitution attack*: The adversary R_L is now allowed to observe some valid message-tag pairs associated with an m-dimensional subspace $V = \text{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$. Clearly, since R_L can read the messages it received, we assume that the messages transiting through the corrupted nodes cannot span a subspace with dimension larger than m. The

goal of R_L is to forge a new massage $\mathbf{v} \notin V$ that is accepted as authentic by R_i with a success probability expressed as:

$$P_S[i, L] = \max_{\mathbf{k}_L \in \mathcal{K}_L} \max_{v' \in \Upsilon', \Lambda} \max_{\overline{v} \in \overline{\Upsilon}^m} P(v' \text{ is valid for } R_i | \overline{v}, \mathbf{k}_L) \tag{4}$$

where Λ is used to denote a condition such that $\mathbf{v}' \notin \text{span}(\overline{\mathbf{v}_1}, \cdots, \overline{\mathbf{v}_m})$ (see above).

3.3 Bounds

On $P_I[i, L], P_{I_{sub}}[i, L], P_S[i, L]$ We are now ready to derive some information-theoretic bounds on the security parameters (success probabilities) for our general MRHA-code scheme as defined in Figure 2.

Theorem 1. *Let $P_I[i, L], P_{I_{sub}}[i, L], P_S[i, L]$ be defined as in Section 3.1, we then have:*

(*i*) $P_I[i, L] \geq 2^{-I(\Upsilon; \mathcal{K}_i | \mathcal{K}_L)}$;
(*ii*) $P_{I_{sub}}[i, L] \geq 2^{-I(\mathcal{K}_i; \overline{\Upsilon}^m | \mathcal{K}_L)}$;
(*iii*) $P_S[i, L] \geq 2^{-I(\Upsilon'; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L)}$.

The proof for the above theorem is rather involved. Generally speaking, for each of the three inequalities, we give the proof in two main steps by extending the basic proof idea from [11] to deal with more parameters and complex scenarios in our setting. To illustrate this, let us take inequality (*i*) as an example:

1. We first prove that $P_I[i, L] \geq 2^{-I(\mathcal{K}_i; \widetilde{\Upsilon} | \mathcal{K}_L)}$: This may not be trivial and requires a combination of techniques from unconditional probability theory, log-sum inequality, mutual information and entropy.
2. We then prove that $I(\mathcal{K}_i; \widetilde{\Upsilon} | \mathcal{K}_L) = I(\mathcal{K}_i; \Upsilon | \mathcal{K}_L)$: This can be done from the definitions of LHS and RHS.

Following the above two steps, we immediately have $P_I[i, L] \geq 2^{-I(\mathcal{K}_i; \Upsilon | \mathcal{K}_L)}$.

We use the similar steps and techniques to prove inequalities (*ii*) and (*iii*), but make more efforts in the second step proofs. The full details of the proof is shown in Appendix A.

On $|\mathcal{K}|$ and $|\mathcal{K}_i| (1 \leq i \leq N)$ Moreover, we present some lower bounds on the efficiency parameters, that is the numbers of keys that are required at the sender and the receivers, in terms of success probabilities in attacking an (q, m, n, N)-MRHA-code $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$. To consider attacks from all possible $(l+1)$-subsets $L \cup i (i \notin L)$ of $\{1, \ldots, N\}$, we let:

$$\overline{P_I} = \max_{L \cup i} \{P_I[i, L]\}; \overline{P_{I_{sub}}} = \max_{L \cup i} \{P_{I_{sub}}[i, L]\}; \overline{P_S} = \max_{L \cup i} \{P_S[i, L]\}.$$

Particularly, we consider $L = \{i_1, \ldots, i_{\mu-1}\}$, that is, there are at most $\mu - 1 (\mu < N)$ corrupted nodes in the network. An (q, m, n, μ, N)-MRHA-code is an (q, m, n, N)-MRHA-code where no subset of $\mu - 1$ verifiers can fool another verifier into accepting a forged message. We capture this more precisely with the following definition.

Definition 3. *An $[M, M_1, \ldots, M_N, d_1, d_2, d_3]$ (q, m, n, μ, N)-MRHA-code is an (q, m, n, N)-MRHA-code $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ where $L = \{i_1, \ldots, i_{\mu-1}\}$ and $|\mathcal{K}| = M, |\mathcal{K}_1| = M_1, \ldots, |\mathcal{K}_N| = M_N, \overline{P_I} = d_1, \overline{P_{I_{sub}}} = d_2, \overline{P_S} = d_3$.*

Suppose γ is a pre-determined value (part of the security parameters), we then have the following theorem for $|\mathcal{K}|$ and $|\mathcal{K}_i|, 1 \le i \le N$.

Theorem 2. *Given an (q, m, n, μ, N)-MRHA-code $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ where $\overline{P_{I_{sub}}} \le (\overline{P_I})^m$ and $\mathsf{max}(\overline{P_I}, \overline{P_S}) \le \frac{1}{\gamma}$, we have:*

(i) $|\mathcal{K}_i| \ge \gamma^n$ *for each* $1 \le i \le N$;
(ii) $|\mathcal{K}| \ge \gamma^{\mu n}$.

To prove the above theorem, we exploit the results from Theorem 1 and combine them with some entropy equalities and inequalities. We also rely on $n-m$ "independent" subspace substitution attacks in our proof.

Proof. *(i)* For each $(\mu-1)$-subset L from $\{1, \ldots, N\}$ and any $i \in \{1, \ldots, N\}, i \notin L$, we consider the attacks from R_L on R_i. More precisely, we consider the case where R_L performs a subspace impersonation attack on a given subspace $V = \mathrm{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$ and $n-m$ "independent" subspace substitution attacks after he has observed a valid tag for V. In the attacks, R_L chooses a forged message-tag pair $(v_i, t_i) = u_i \in \Upsilon'_i$ for the i-th attack, where $\Upsilon'_i = \mathrm{span}(u_i)$ with $v_i \notin \mathrm{span}(\mathbf{v}_1, \cdots, \mathbf{v}_m)$ by the definition of a subspace substitution attack. We say the $n-m$ subspace substitution attacks are independent if it holds that

$$\Upsilon'_1 \ne \Upsilon'_2 \cdots \ne \Upsilon'_{n-m-1} \ne \Upsilon'_{n-m}.$$

Since V is an m-dimensional subspace of the source space $\mathcal{S}$, which in turn is an $n-$dimensional vector space (over $\mathbb{F}_q$), it is reasonable to assume that R_L conducts the $n-m$ independent subspace substitution attacks after observing a tag for V.

With the above consideration, assuming that $\overline{P_{I_{sub}}} \le (\overline{P_I})^m$ and $\mathsf{max}(\overline{P_I}, \overline{P_S}) \le \frac{1}{\gamma}$, and using the results from Theorem 1, we have:

$$\begin{aligned}(\frac{1}{\gamma})^n &\ge (\overline{P_I})^m (\overline{P_S})^{n-m} \ge \overline{P_{I_{sub}}} (\overline{P_S})^{n-m} \ge P_{I_{sub}}[i, L](P_S[i, L])^{n-m} \\ &\ge 2^{-I(\mathcal{K}_i; \overline{\Upsilon}^m | \mathcal{K}_L)} 2^{-(I(\Upsilon'_1; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + I(\Upsilon'_{n-m}; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L))} \end{aligned} \tag{5}$$

Now it is sufficient to prove the inequality below:

$$I(\Upsilon'_1; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + I(\Upsilon'_{n-m}; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L) \le H(\mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L). \tag{6}$$

Since if (6) is proved, from (5), we have:

$$\begin{aligned}(\frac{1}{\gamma})^n &\ge 2^{-(I(\mathcal{K}_i; \overline{\Upsilon}^m | \mathcal{K}_L) + I(\Upsilon'_1; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + I(\Upsilon'_{n-m}; \mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L))} \\ &\ge 2^{-(I(\mathcal{K}_i; \overline{\Upsilon}^m | \mathcal{K}_L) + H(\mathcal{K}_i | \overline{\Upsilon}^m, \mathcal{K}_L))} = 2^{-H(\mathcal{K}_i | \mathcal{K}_L)} \\ &\ge 2^{-H(\mathcal{K}_i)} \ge 2^{-\log |\mathcal{K}_i|} = \frac{1}{|\mathcal{K}_i|}\end{aligned}$$

which implies that $|\mathcal{K}_i| \ge \gamma^n$, and hence part (i) of our theorem is proved.

Indeed, we can prove (6) as follows:

$$\begin{aligned}
&H(\mathcal{K}_i) = I(\mathcal{K}_i; \Upsilon'_1, \cdots, \Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L) + H(\mathcal{K}_i|\Upsilon'_1, \cdots, \Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L) \Rightarrow \\
&H(\mathcal{K}_i) \geq I(\mathcal{K}_i; \Upsilon'_1, \cdots, \Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L) \Rightarrow \\
&H(\mathcal{K}_i) \geq \sum_{i=1}^{n-m} I(\mathcal{K}_i; \Upsilon'_i, \overline{\Upsilon}^m, \mathcal{K}_L|\Upsilon'_{i-1}, \Upsilon'_{i-2}, \cdots, \Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L) \Rightarrow \\
&H(\mathcal{K}_i) \geq I(\mathcal{K}_i; \Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + I(\mathcal{K}_i; \Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L) \Rightarrow \\
&(n-m-1)H(\mathcal{K}_i) \leq (n-m)H(\mathcal{K}_i) - (I(\mathcal{K}_i; \Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + I(\mathcal{K}_i; \Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L)) \Rightarrow \\
&(n-m-1)H(\mathcal{K}_i) \leq (H(\mathcal{K}_i) - I(\mathcal{K}_i; \Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L)) + \cdots + (H(\mathcal{K}_i) - I(\mathcal{K}_i; \Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L)) \Rightarrow \\
&(n-m-1)H(\mathcal{K}_i) \leq H(\mathcal{K}_i|\Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + H(\mathcal{K}_i|\Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L). \qquad (7)
\end{aligned}$$

The transition in (7) is due to the facts that $n > m$ and $H(\mathcal{K}_i) \geq H(\mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L) \geq 0$. Since (7) implies

$$(n-m-1)H(\mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L) \leq H(\mathcal{K}_i|\Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + H(\mathcal{K}_i|\Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L)$$

which in turn implies

$$\begin{aligned}
&H(\mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L) - H(\mathcal{K}_i|\Upsilon'_1, \overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + H(\mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L) - H(\mathcal{K}_i|\Upsilon'_{n-m}, \overline{\Upsilon}^m, \mathcal{K}_L) \\
&\leq H(\mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L).
\end{aligned}$$

Hence, we have

$$I(\Upsilon'_1; \mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L) + \cdots + I(\Upsilon'_{n-m}; \mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L) \leq H(\mathcal{K}_i|\overline{\Upsilon}^m, \mathcal{K}_L)$$

as required.

(*ii*) Assume that $L_i = \{1, \ldots, i-1, i+1, \ldots, \mu\}$ for $1 \leq i \leq \mu$, we then have:

$$\begin{aligned}
(\frac{1}{\gamma})^{\mu n} &\geq \prod_{i=1}^{\mu} (\overline{P_I})^m (\overline{P_S})^{n-m} \geq \prod_{i=1}^{\mu} \overline{P_{I_{sub}}} (\overline{P_S})^{n-m} \geq \prod_{i=1}^{\mu} P_{I_{sub}}[i, L](P_S[i, L])^{n-m} \\
&\geq 2^{\sum_{i=1}^{\mu} -H(\mathcal{K}_i|\mathcal{K}_{L_i})} \geq 2^{-\sum_{i=1}^{\mu} H(\mathcal{K}_i|\mathcal{K}_1, \ldots, \mathcal{K}_{i-1})} = 2^{-H(\mathcal{K}_1, \ldots, \mathcal{K}_\mu)} \\
&\geq 2^{-H(\mathcal{K})} \geq 2^{-log|\mathcal{K}|} = \frac{1}{|\mathcal{K}|}.
\end{aligned}$$

Therefore, $|\mathcal{K}| \geq \gamma^{\mu n}$, as required.

□

The bounds derived in Theorem 2 are tight. In what follows, we give a construction meeting these bounds.

4 Constructions

Construction of an MRHA-code involves choosing N homomorphic A-codes $\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N$ and defining two mappings τ_i, π_i for each $1 \leq i \leq N$. We make use of previous works on optimal A-codes construction [11, 15] and we work on a polynomial-based secret sharing scheme [12].

4.1 MRHA-code Schemes

We derive a class of MRHA-codes from a class of homomorphic A-codes $\mathcal{C}_0$ (see below). For each of the MRHA-code, we then construct an MRHA-code scheme. We illustrate this below.

Given an (q, m, n)-homomorphic A-code $\mathcal{C}_0 = (\mathcal{S}, \mathcal{K}, \mathcal{A}, f)$ where $\mathcal{A} = \mathcal{S}$ (that is, $t = n$), f is a natural mapping: $f(\mathbf{v}, \mathbf{k}) = \mathbf{vk}$ for any $\mathbf{v} \in \mathcal{S}$ and $\mathbf{k} \in \mathcal{K}$. Moreover, $\mathcal{K} \subseteq \mathbb{F}_q^{n\times n}$ is a subspace over $\mathbb{F}_q$ and satisfies *Property 1*: $AB \in \mathcal{K}$ holds for any $B \in \mathcal{K}$ and any non-singular matrix $A \in \mathbb{F}_q^{n\times n}$. From now on, we assume that $N < q^n$ and take this A-code $\mathcal{C}_0$ as a building block to construct an (q, m, n, N)-MRHA-code $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ as follows:[3]

- $\mathcal{C} = (\mathcal{S}, \overline{\mathcal{K}}, \overline{\mathcal{A}}, \overline{f})$, where $\overline{\mathcal{K}} = \mathcal{K} \times \cdots \times \mathcal{K}$ is the collection of all μ-tuple elements, each of which is from $\mathcal{K}$; $\overline{\mathcal{A}} = \mathcal{A} \times \cdots \times \mathcal{A}$ is the collection of all μ-tuple elements, each of which is from $\mathcal{A}$. For any $\mathbf{v} \in \mathcal{S}$ and $(\overline{\mathbf{k}_0}, \ldots, \overline{\mathbf{k}}_{\mu-1}) = \overline{\mathbf{k}} \in \overline{\mathcal{K}}$, the mapping $\overline{f}$ is defined as:
$$\overline{f}(\mathbf{v}, \overline{\mathbf{k}}) = (\mathbf{v}\overline{\mathbf{k}}_0, \ldots, \mathbf{v}\overline{\mathbf{k}}_{\mu-1}).$$
- We randomly choose a nonzero element $x_j \in \mathbb{F}_{q^n}$ for each $1 \le j \le N$ such that $x_h \neq x_l$ if $h \neq l$.[4] The mapping $\tau_j : \overline{\mathcal{K}} \to \mathcal{K}_j$ is then defined as:
$$\tau_j(\overline{\mathbf{k}}) = \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i.$$
Writing $\overline{\mathbf{t}} \in \overline{\mathcal{A}}$ as $\overline{\mathbf{t}} = (\overline{\mathbf{t}}_0, \ldots, \overline{\mathbf{t}}_{\mu-1})$, the mapping $\pi_j : \overline{\mathcal{A}} \to \mathcal{A}_j$ is constructed via:
$$\pi_j(\overline{\mathbf{t}}) = \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{t}}_i.$$
- For each $1 \le j \le N$, we set $\mathcal{C}_j = (\mathcal{S}, \mathcal{K}_j, \mathcal{A}_j, f_j)$, where $\mathcal{K}_j = \mathcal{K}$ and $\mathcal{A}_j = \mathcal{A}$. For any $\mathbf{v} \in \mathcal{S}, \mathbf{k} \in \mathcal{K}_j$, the mapping f_j is defined simply as $f_j(\mathbf{v}, \mathbf{k}) = \mathbf{vk}$.

With these settings, it is easy to check that for each $1 \le j \le N$, π_j is $\mathbb{F}_q$-linear and for any $(\mathbf{v}, \overline{\mathbf{k}}) \in \mathcal{S} \times \overline{\mathcal{K}}$ we have
$$f_j(\mathbf{v}, \tau_j(\overline{\mathbf{k}})) = \mathbf{v} \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = \sum_{i=0}^{\mu-1} x_j^i \mathbf{v}\overline{\mathbf{k}}_i = \pi_j(f(\mathbf{v}, \overline{\mathbf{k}})),$$
which implies that the constructed A-code $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ is an (q, m, n, N)-MRHA-code.

With the A-code, we have a message authentication scheme as shown in Figure 3.

It is a known fact that for any $v \in \mathbb{F}_{q^n}$ and $B \in \mathbb{F}_q^{n\times n}$, we have $vB = M_v B$ where the non-singular matrix $M_v \in \mathbb{F}_q^{n\times n}$ is the multiplication matrix for v when fixing an $\mathbb{F}_q$-basis of $\mathbb{F}_{q^n}$. Combing this fact with our assumption that $\mathcal{K}$ is a subspace over $\mathbb{F}_q$ and satisfies Property 1, we have $P(j) = \tau_j(\overline{\mathbf{k}}) \in \mathcal{K}$ for any $1 \le j \le N$ and $\overline{\mathbf{k}} \in \overline{\mathcal{K}}$. This ensures that the mapping f_j is well defined for all $1 \le j \le N$. Moreover, in our construction, the values x_j are public for $1 \le j \le N$ and do not have impact on the security of our scheme, as analyzed in the next subsection.

[3] In fact, we require some computations rules for our construction. We fix an $\mathbb{F}_q$-basis of $\mathbb{F}_{q^n}$, say $(\nu_1, \ldots, \nu_n)$. For any $\alpha \in \mathbb{F}_{q^n}$ and any $(v_1, \ldots, v_n) = v \in \mathbb{F}_q^n$, we compute αv (or $v\alpha$) by considering v as an element from $\mathbb{F}_{q^n}$ (that is $v' = \sum_{i=1}^n v_i\nu_i$), and then do multiplication $\alpha v'$ (or $v'\alpha$) in $\mathbb{F}_{q^n}$. For any $\alpha \in \mathbb{F}_{q^n}$ and any $k \in \mathbb{F}_q^{n\times n}$, we compute αk by expressing α as a vector $\alpha' = (\alpha'_1, \ldots, \alpha'_n) \in \mathbb{F}_q^n$ with $\sum_{i=1}^n \alpha'_i\nu_i = \alpha$ and then do matrix operation $\alpha' k$ in $\mathbb{F}_q$.

[4] That is why we require $N < q^n$.

– **Key distribution:** A trusted authority (or the source itself) randomly chooses a private key $(\overline{\mathbf{k}}_0, \ldots \overline{\mathbf{k}}_{\mu-1}) = \overline{\mathbf{k}} \in \overline{\mathcal{K}}$ for the source. For each verifier R_j (for $1 \le j \le N$), the trusted authority computes and sends R_j's private key as $P(j) = \tau_j(\overline{\mathbf{k}}) = \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i$.
– **Tag generation:** For each message $\mathbf{v}_i \in \mathbb{F}_q^n$ $(1 \le i \le m)$, the source computes $(\overline{\mathbf{t}}_{i0}, \ldots, \overline{\mathbf{t}}_{i,\mu-1}) = (\mathbf{v}_i\overline{\mathbf{k}}_0, \ldots, \mathbf{v}_i\overline{\mathbf{k}}_{\mu-1})$ as the corresponding tag $\overline{\mathbf{t}}_i$ and sends out the message-tag pair $(\mathbf{v}_i, \overline{\mathbf{t}}_i)$.
– **Combination:** Assume that each intermediate node receives some message-tag pair $(\mathbf{x}_h, \overline{\mathbf{t}}_{\mathbf{x}_h})$ for some index h, where each $\mathbf{x}_h$ is already a linear combination of some messages $\mathbf{v}_i$. The intermediate node computes $(\overline{\mathbf{t}}_{\mathbf{y}0}, \ldots, \overline{\mathbf{t}}_{\mathbf{y},\mu-1})$ as the tag $\overline{\mathbf{t}}_{\mathbf{y}}$ corresponding to an output vector $\mathbf{y} = \sum_h \alpha_h \mathbf{x}_h$, where $\overline{\mathbf{t}}_{\mathbf{y}l} = \sum_h \alpha_h \mathbf{t}_{\mathbf{x}_h l}$ and the sum is taken over some subset of the received tags.
– **Verification:** Assume that a verifier R_j possesses a private key $P(j)$ (together with the associated public component x_j) and it receives a message $\mathbf{v}$ and the corresponding tag $\overline{\mathbf{t}} = (\overline{\mathbf{t}}_{\mathbf{v}0}, \ldots, \overline{\mathbf{t}}_{\mathbf{v},\mu-1})$. The verifier checks if $\mathbf{v}P(j) = \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{t}}_{\mathbf{v}i}$; it accepts $\mathbf{v}$ as authentic if the equation holds; otherwise it rejects.

Fig. 3. Construction of MRHA-code scheme.

4.2 Security Analysis

We analyze the security of the class of A-codes $(\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ constructed in Section 4.1 against an arbitrary subset of $\mu - 1$ corrupted nodes. Without loss of generality, we denote the corrupted nodes as $R_L = (R_1, \ldots, R_{\mu-1})$ and the honest node as R_μ. When we write $\overline{\mathcal{K}^\mu} \subseteq \overline{\mathcal{K}}$, we also mean

$$\overline{\mathcal{K}^\mu} = \{\overline{\mathbf{k}} \in \overline{\mathcal{K}} | \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j), 1 \le j \le \mu - 1\}.$$

We assume that all the messages and private keys are uniformly distributed. Since the sets of R_L and R_μ are arbitrary chosen, according to (2), (3), (4) and the probabilities $\overline{P_I}, \overline{P_{I_{sub}}}, \overline{P_S}$ defined in Section 3.3, we have (recall that $V = \mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_m)$):

$$\overline{P_I} = \max_{\mathbf{0} \ne \mathbf{v} \in \mathcal{S}, \overline{\mathbf{t}}_{\mathbf{v}i}} \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu}| \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{t}}_{\mathbf{v}i} = \mathbf{v}P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|};$$

$$\overline{P_{I_{sub}}} = \max_{V, \overline{\mathbf{t}}_{\mathbf{v}_j i}} \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu}| \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{t}}_{\mathbf{v}_j i} = \mathbf{v}_j P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|};$$

$$\overline{P_S} = \max_{V, \mathbf{v} \notin V, \overline{\mathbf{t}}_{\mathbf{v}_j i}, \overline{\mathbf{t}}_{\mathbf{v}i}} \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j\overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}, \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{t}}_{\mathbf{v}i} = \mathbf{v}P(\mu)\}|}{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j\overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}\}|}.$$

Definition 4. *(Definition 2 of [13]) An (q, m, n)-homomorphic A-code $(\mathcal{S}, \mathcal{A}, \mathcal{K}, f)$ is called an $[M, d_1, d_2, d_3]$ (q, m, n)-homomorphic A-code, if $|\mathcal{K}| = M, P_I = d_1, P_{I_{sub}} = d_2$ and $P_S = d_3$.*

From the above assumptions and Definition 4, which covers both the relevant security and efficiency parameters, we derive the following theorem.

Theorem 3. *If the original A-code $\mathcal{C}_0 = (\mathcal{S}, \mathcal{A}, \mathcal{K}, f)$ is an $[M, d_1, d_2, d_3]$ (q, m, n)-homomorphic A-code, then the A-code constructed in Section 4.1 is an $[M^\mu, M, \ldots, M, d_1, d_2, d_3]$ (q, m, n, μ, N)-MRHA-code and requires μ tags appended per message in network coding.*

The proof for Theorem 3 is presented in Appendix B. Theorem 3 implies that a collusion of $\mu - 1$ malicious nodes gives the same probabilities of successful attacks as an outside attacker, in terms of cheating the μ-th node. Our HMRA-codes, therefore, achieve the same security as

the the trivial scheme described in Section 1 (when the number of corrupted nodes involved in a collusion is $\leq \mu - 1$). However, clearly, our construction is more efficient than the trivial scheme. In the latter, the sender needs to store N keys and generate/transmit N tags per message. On the other hand, our scheme requires $\mu < N$ keys at the sender and μ tag for each message. Nevertheless, we note that each verifier is required to store the same number of keys in both schemes.

Particularly, we now borrow a concrete example homomorphic A-code from [13] and use it as a building block to construct an MRHA-code which meets the key size bound from Theorem 2.

Example 1. Let $A \in \mathbb{F}_q^{n \times n}$ be a matrix with $\mathsf{rank}(A) = d$. We use $\langle A \rangle_R$ to denote the row space of A, namely, the set of all possible $\mathbb{F}_q$-linear combinations of its row vectors. We denote $\mathcal{K}_A$ as the set of all matrices B_A where $B_A \in \mathbb{F}_q^{n \times n}$ and each row of B_A belongs to $\langle A \rangle_R$.

Theorem 4. *Given a matrix $A \in \mathbb{F}_q^{n \times n}$, where $\mathsf{rank}(A) = d$ and an (q, m, n)-homomorphic A-code $\mathcal{C}_0 = (\mathcal{S}, \mathcal{A}, \mathcal{K}, f)$, where $\mathcal{K} = \mathcal{K}_A$ as defined in Example 1, we construct an A-code $\overline{\mathcal{C}} = (\mathcal{C}, \mathcal{C}_1, \ldots, \mathcal{C}_N)$ as shown in Section 4.1. The A-code $\overline{\mathcal{C}}$ is an $[q^{\mu nd}, q^{nd}, \ldots, q^{nd}, q^{-d}, q^{-md}, q^{-d}]$ (q, m, n, μ, N)-MRHA-code for all q, m, n, d, N with $m < n$, $1 \leq d \leq n$ and $N < q^n$.*

Proof. Firstly, one can easily check that $\mathcal{K} = \mathcal{K}_A$ in $\mathcal{C}_0 = (\mathcal{S}, \mathcal{A}, \mathcal{K}, f)$ is a subspace over $\mathbb{F}_q$ and satisfies Property 1. Since $N < q^n$, one can always construct $\overline{\mathcal{C}}$ from $\mathcal{C}_0$ in a way as shown in Section 4.1. Secondly, Theorem 3 in [13] guarantees that $\mathcal{C}_0$ is an $[q^{nd}, q^{-d}, q^{-md}, q^{-d}]$ (q, m, n)-homomorphic A-code for all q, m, n, d with $m < n$ and $1 \leq d \leq n$. With these and our Theorem 3, Theorem 4 follows.

Comparing Theorem 4 with the bounds from Theorem 2, it is easy to deduce the following corollary.

Corollary 1. *Both the security and efficiency parameters in Theorem 4 meet the bounds specified in Theorem 2.*

5 Conclusions and Open Problems

We have given a formal definition of an MRHA-code, derived some bounds on their security and efficiency, as well as given some efficient constructions. However, numerous challenging open problems remain.

One natural next step is to investigate MRHA-codes for a dynamic sender. It is not at all clear if our current results can be applied directly to cope with a dynamic sender. Moreover, a more challenging problem is to consider a setting where there exist multiple senders in the network. Particularly, in the so-called *inter-session* network coding setting, where messages from different sources are transmitted to and (possibly) mixed by an intermediate node, our MRHA-codes would not be able to prevent pollution attacks. We now have to consider not only malicious intermediate nodes, but also malicious sources.

References

1. S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-Based Integrity for Network Coding", *ACNS* 2009, pp.292-305, Springer Verlag, 2009.

2. A. Apavatjrut, W. Znaidi, A. Fraboulet, C. Goursaud, K. Jaffrés-Runser, C. Lauradoux, and M. Minier. "Energy efficient authentication strategies for network coding". *Concurrency and Computation: Practice and Experience*, 2011.
3. D. Boneh, D. Freeman, J. Katz, B. Water, "Signing a Linear Subspace: Signature Schemes for Network Coding", *Public Key Cryptography, PKC* 2009, vol 5443 of LNCS, pp. 68 – 87, Springer Verlag, 2009.
4. L. Carter and M. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, vol. 18, pp.143-154, 1979.
5. D. Charles, K. Jain, K. Lauter, "Signatures for Network Coding",*40th Annual Conference on Information Sciences and Systems*, 2006. Available at http://eprint.iacr.org.
6. C. Fragouli and E. Soljanin, "Network Coding Fundamentals", *Now Publishers Inc*, 2007.
7. D. M. Freeman. "Improved security for linearly homomorphic signatures: A generic framework". *Public Key Cryptography, PKC* 2012, volume 7293 of LNCS, pp. 697-714, Springer Verlag, 2012.
8. Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "RIPPLE Authentication for Network Coding," *INFOCOM* 2010, 2010.
9. E. N. Gilbert, F. J MacWilliams, N. J. A. Sloane, "Codes which detect deception," *Bell Syst. Techn. Journal*, vol. 33, 1974.
10. F. Oggier and H. Fathi, "An Authentication Code against Pollution Attacks in Network Coding", *IEEE/ACM Transactions on Networking*, Issue 99, March 2011. CoRR abs/0909.3146, 2009.
11. R. Safavi-Naini, H. Wang," Multi-receiver Authentication Codes: Models, Bounds, Constructions, and Extensions", *Inf. Comput. (IANDC)*, vol. 151, pp.148-172, 1999.
12. A.Shamir, "How to share a secret", *Commun. ACM,* vol. 22, pp. 612-613, 1979.
13. Z. Tang, "Homomorphic A-codes for Network Coding", ePrint Archive, http://eprint.iacr.org/2012/331.
14. M.Walker, Information-Theoretic Bounds for Authentication Schemes, *J. of Cryptology*, Vol 2, No. 3, pp. 131-143, 1990.
15. H. Wang, C. Xing and R. Safavi-Naini, "Linear Authentication Codes: Bounds and Constructions," *IEEE Trans. on Information Theory*, vol. 49, no. 4, 2003.

A Proof of Theorem 1

(*i*) $P_I[i,L] \geq 2^{-I(\Upsilon;\mathcal{K}_i|\mathcal{K}_L)}$

– On one hand, from the definition of $P_I[i,L]$, we have:

$$P_I[i,L] = \max_{\substack{\upsilon\in\widetilde{\Upsilon}\\ \mathbf{k}_L\in\mathcal{K}_L}} \{P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L\in\mathcal{K}_L)\} \geq \sum_{\mathbf{k}_L\in\mathcal{K}_L} P(\mathbf{k}_L)\big[\max_{\upsilon\in\widetilde{\Upsilon}} P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L)\big]$$

$$\geq \sum_{\mathbf{k}_L\in\mathcal{K}_L} P(\mathbf{k}_L)\big[\sum_{\upsilon\in\widetilde{\Upsilon}} P(\upsilon|\mathbf{k}_L)P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L)\big].$$

By log-sum inequality, we furthermore have:

$$\log P_I[i,L] \geq \sum_{\mathbf{k}_L\in\mathcal{K}_L} P(\mathbf{k}_L)\big[\sum_{\upsilon\in\widetilde{\Upsilon}} P(\upsilon|\mathbf{k}_L)\log P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L)\big] \tag{8}$$

– On the other hand, we firstly define a characteristic function which will be used later, that is, a function $\mathcal{X}_I(\upsilon,\mathbf{k})$ on $\widetilde{\Upsilon}\times\mathcal{K}_i\times\mathcal{K}_L$ by:

$$\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L) = \begin{cases} 1 & \text{if there exists a } \mathbf{k}\in\mathcal{K} \text{ such that } \mathbf{t}{=}\mathbf{vk},\ \tau_i(\mathbf{k})=\mathbf{k}_i, \tau_L(\mathbf{k})=\mathbf{k}_L;\\ 0 & \text{otherwise.}\end{cases}$$

Obviously, $\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)=1$ implies that $f_i(\mathbf{v},\tau_i(\mathbf{k})) = f_i(\mathbf{v},\mathbf{k}_i) = \pi_i(f(\mathbf{v},\mathbf{k})) = \pi_i(\mathbf{t})$ when provided $\mathbf{k}_L$, and thus implies that $\upsilon=(\mathbf{v},\mathbf{t})$ is valid for R_i when provided $\mathbf{k}_L$.

With this, we compute $I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L)$ as:

$$I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L) = E_{P(\upsilon,\mathbf{k}_i,\mathbf{k}_L)}\frac{P(\widetilde{\Upsilon},\mathcal{K}_i|\mathcal{K}_L)}{P(\widetilde{\Upsilon}|\mathcal{K}_L)P(\mathcal{K}_i|\mathcal{K}_L)} = \sum_{\substack{\upsilon\in\widetilde{\Upsilon},\mathbf{k}_i\in\mathcal{K}_i\\ \mathbf{k}_L\in\mathcal{K}_L}} P(\upsilon,\mathbf{k}_i,\mathbf{k}_L)\log\frac{P(\upsilon,\mathbf{k}_i|\mathbf{k}_L)}{P(\upsilon|\mathbf{k}_L)P(\mathbf{k}_i|\mathbf{k}_L)}$$

$$= \sum_{\upsilon\in\widetilde{\Upsilon},\mathbf{k}_i\in\mathcal{K}_i,\mathbf{k}_L\in\mathcal{K}_L} P(\upsilon,\mathbf{k}_i,\mathbf{k}_L)\log\frac{P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)P(\upsilon|\mathbf{k}_L)}{P(\upsilon|\mathbf{k}_L)P(\mathbf{k}_i|\mathbf{k}_L)}$$

$$= \sum_{\upsilon\in\widetilde{\Upsilon},\mathbf{k}_L\in\mathcal{K}_L,P(\upsilon,\mathbf{k}_L)\neq 0} P(\upsilon,\mathbf{k}_L)\big(\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\log\frac{P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)}{P(\mathbf{k}_i|\mathbf{k}_L)}\big).$$

We note here that, for each pair $(\upsilon,\mathbf{k}_L)$, when $P(\upsilon,\mathbf{k}_L)\neq 0$, if $\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)=0$, then $P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)=0$; in this case, $P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\log\frac{P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)}{P(\mathbf{k}_i|\mathbf{k}_L)}=0$. So, the summation taken over $\mathcal{K}$ above is restricted to all $\mathbf{k}_i$ for which $\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)=1$. Henceforth, we have:

$$I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_l) = \sum_{\substack{\upsilon\in\widetilde{\Upsilon},\mathbf{k}_L\in\mathcal{K}_L\\ P(\upsilon,\mathbf{k}_L)\neq 0}} P(\upsilon,\mathbf{k}_L)\big(\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)\log\frac{P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)}{P(\mathbf{k}_i|\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)}\big).$$

By log-sum inequality, we furthermore have:

$$I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L) \geq \sum_{\upsilon\in\widetilde{\Upsilon},\mathbf{k}_L\in\mathcal{K}_L,P(\upsilon,\mathbf{k}_L)\neq 0} P(\upsilon,\mathbf{k}_L)\big(\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)\big)$$
$$\times\log\frac{\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)}{\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)}. \tag{9}$$

Again, as we have observed, if $P(\upsilon,\mathbf{k}_L)\neq 0$ and $\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)=0$, then $P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)=0$. This implies

$$\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon,\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L)=1 \tag{10}$$

and

$$\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\mathbf{k}_L)\mathcal{X}_I(\upsilon,\mathbf{k}_i,\mathbf{k}_L) = P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L). \tag{11}$$

Based on (10) and (11) above, we continue (9) and have:

$$\begin{aligned} I(\mathcal{K};\widetilde{\Upsilon}|\mathcal{K}_L) \geq & \quad -\sum_{\upsilon\in\widetilde{\Upsilon},\mathbf{k}_L\in\mathcal{K}_L} P(\upsilon,\mathbf{k}_L)\log P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L) \\ = & \quad -\sum_{\mathbf{k}_L\in\mathcal{K}_L} P(\mathbf{k}_L)\big[\sum_{\upsilon\in\widetilde{\Upsilon}} P(\upsilon|\mathbf{k}_L)\log P(\upsilon \text{ is valid for } R_i|\mathbf{k}_L)\big]. \end{aligned} \tag{12}$$

Now, we consider (12) together with (8) and have:

$$P_I[i,L]\geq 2^{-I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L)}.$$

Furthermore, it is obvious to see that $I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L) = I(\mathcal{K}_i;\Upsilon|\mathcal{K}_L)$. Indeed, if we write $I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L)$ as $I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L) = I(\mathcal{K}_i;\Upsilon|\mathcal{K}_L) + I(\mathcal{K}_i;\widetilde{\Upsilon}'|\mathcal{K}_L)$ where $\widetilde{\Upsilon}' = \widetilde{\Upsilon}\backslash\Upsilon$, then it holds trivially that $I(\mathcal{K}_i;\widetilde{\Upsilon}'|\mathcal{K}_L) = 0$.
Finally, we have $P_I[i,L] \geq 2^{-I(\mathcal{K}_i;\widetilde{\Upsilon}|\mathcal{K}_L)} = 2^{-I(\mathcal{K}_i;\Upsilon|\mathcal{K}_L)}$ and thus complete the proof of (i) in the theorem.

(ii) $P_{I_{sub}}[i,L] \geq 2^{-I(\mathcal{K}_i;\overline{\Upsilon}^m|\mathcal{K}_L)}$

We denote $\Upsilon^m = \Upsilon \times \cdots \times \Upsilon$ as the collection of all m-tuple elements, each of which is from Υ; formally, $\Upsilon^m = \{(\upsilon_1, \ldots, \upsilon_m) : \upsilon_i \in \Upsilon, 1 \leq i \leq m\}$. We can prove the inequality (13) below, in a quite similar way as the proof above for (i) (we don't repeat the proof details here due to space constraints.):

$$P_{I_{sub}}[i,L] \geq 2^{-I(\mathcal{K}_i;\Upsilon^m|\mathcal{K}_L)}. \tag{13}$$

Furthermore, we claim that

$$I(\mathcal{K}_i;\Upsilon^m|\mathcal{K}_L) = I(\mathcal{K}_i;\overline{\Upsilon}^m|\mathcal{K}_L). \tag{14}$$

Indeed, if we write $\underline{\Upsilon^m} = \Upsilon^m\backslash\overline{\Upsilon}^m$, we can prove that all the mutual information between $\mathcal{K}_i$ and $\underline{\Upsilon^m}$ when provided $\mathcal{K}_L$ is included in the mutual information between $\mathcal{K}_i$ and $\overline{\Upsilon}^m$ when provided $\mathcal{K}_L$, which proves our claim (14). We demonstrate the claim by investigating a random sequence $(y_1, \cdots, y_m) = y \in \underline{\Upsilon^m}$ and a random key $\mathbf{k}_L \in \mathcal{K}_L$. For any $v \in \Upsilon$ below, we denote it as $v = (\mathbf{v}_v, \mathbf{t}_v)$ showing $\mathbf{v}_v$ is the message part while $\mathbf{t}_v$ is the tag part. There are two cases:

Case 1: if there is any key $\mathbf{k} \in \mathcal{K}$ such that $f_i(\mathbf{v}_{y_i}, \tau_i(\mathbf{k})) = \pi_i(\mathbf{t}_{y_i})(1 \leq i \leq m)$ and $\tau_L(\mathbf{k}) = \mathbf{k}_L$. (in this case, y_i's are not all linearly independent, which causes $y \notin \overline{\Upsilon}^m$); then in this case there always exists a sequence $(x_1, \cdots, x_m) = x \in \overline{\Upsilon}^m$ such that $f_i(\mathbf{v}_{x_i}, \tau_i(\mathbf{k})) = \pi_i(\mathbf{t}_{x_i})(1 \leq i \leq m)$ and $\tau_L(\mathbf{k}) = \mathbf{k}_L$; therefore the information contained in y which is useful for R_L owning $\mathbf{k}_L$ to disclose the key $\mathbf{k}$, can be considered as a proper subset of and thus included in the useful information contained in x; in other words, from additional y, R_L cannot have more information helpful for disclosing $\mathbf{k}$ than from only x.
Case 2: we consider a proper subset of y, say, $y_{i_1}, \cdots, y_{i_j}, j < m$ and assume there exists any key $\mathbf{k} \in \mathcal{K}$ such that $f_i(\mathbf{v}_{y_i}, \tau_{i_h}(\mathbf{k})) = \pi_i(\mathbf{t}_{y_{i_h}})(1 \leq h \leq j)$ and $\tau_L(\mathbf{k}) = \mathbf{k}_L$; in this case, we can always find a sequence $(x_1, \cdots, x_m) = x \in \overline{\Upsilon}^m$ such $f_i(\mathbf{v}_{x_i}, \tau_i(\mathbf{k})) = \pi_i(\mathbf{t}_{x_i})(1 \leq i \leq m), \tau_L(\mathbf{k}) = \mathbf{k}_L$ and $y_{i_h} \in \text{span}(x_1, \cdots, x_m)(1 \leq h \leq j)$; therefore the information contained in y which can help R_L owning $\mathbf{k}_L$ reveal the key $\mathbf{k}$, can be regarded as a proper subset of and thus included in the helpful information contained in x; in other words, when from this additional y, the adversary cannot have more information helpful for disclosing $\mathbf{k}$ than from only x.

Since the above analysis can be generalized into all sequences $y \in \underline{\Upsilon^m}$ and all keys $\mathbf{k}_L \in \mathcal{K}_L$, we henceforth demonstrate that all the mutual information between $\mathcal{K}_i$ and $\underline{\Upsilon^m}$ when provided $\mathcal{K}_L$ is included in the mutual information between $\mathcal{K}_i$ and $\overline{\Upsilon}^m$ when provided $\mathcal{K}_L$.
Combining (13) and (14), we finally have $P_{I_{sub}}[i,L] \geq 2^{-I(\mathcal{K}_i;\overline{\Upsilon}^m|\mathcal{K}_L)}$ and thus complete the proof of (ii) in the theorem.

(iii) $P_S[i,L] \geq 2^{-I(\Upsilon';\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L)}$

– On one hand, According to the definition of $P_S[i,L]$ from (4), we have:

$$\begin{aligned}
P_S[i,L] &= \max_{\mathbf{k}_L\in\mathcal{K}_L}\max_{\upsilon'\in\Upsilon',\Lambda}\max_{\overline{\upsilon}\in\overline{\Upsilon}^m} P(\upsilon' \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L)\\
&\geq \sum_{\mathbf{k}_L\in\mathcal{K}_L} P(\mathbf{k}_L)\sum_{\overline{\upsilon}\in\overline{\Upsilon}^m} P(\overline{\upsilon}|\mathbf{k}_L)\sum_{\upsilon'\in\Upsilon',\Lambda} P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\upsilon' \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L)\\
&\geq \sum_{\mathbf{k}_L\in\mathcal{K}_L,\overline{\upsilon}\in\overline{\Upsilon}^m} P(\overline{\upsilon},\mathbf{k}_L)\big[\sum_{\upsilon'\in\Upsilon',\Lambda} P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\upsilon' \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L)\big].
\end{aligned}$$

By log-sum inequality, we furthermore have:

$$\log P_S[i,L] \geq \sum_{\substack{\mathbf{k}_L\in\mathcal{K}_L\\ \overline{\upsilon}\in\overline{\Upsilon}^m}} P(\overline{\upsilon},\mathbf{k}_L)\big[\sum_{\upsilon'\in\Upsilon',\Lambda} P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)\log P(\upsilon' \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L)\big]. \tag{15}$$

– On the other hand, we firstly define a characteristic function which will be used later, that is, a function $\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k})$ on $\Upsilon'\times\overline{\Upsilon}^m\times\mathcal{K}_i\times\mathcal{K}_L$ by :

$$\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)=\begin{cases}1 & \text{if there exists a } \mathbf{k}\in\mathcal{K} \text{ such that } \mathbf{t}'=\mathbf{v}'\mathbf{k},\\ & \overline{\mathbf{t}}_i=\overline{\mathbf{v}}_i\mathbf{k}(1\leq i\leq m),\tau_i(\mathbf{k})=\mathbf{k}_i,\tau_L(\mathbf{k})=\mathbf{k}_L;\\ 0 & \text{otherwise.}\end{cases}$$

Obviously, $\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)=1$ implies that $f_i(\mathbf{v}',\tau_i(\mathbf{k}))=f_i(\mathbf{v}',\mathbf{k}_i)=\pi_i(f(\mathbf{v}',\mathbf{k}))=\pi_i(\mathbf{t}')$ when provided $\overline{\upsilon}$ and $\mathbf{k}_L$, and thus implies that $\upsilon'=(\mathbf{v}',\mathbf{t}')$ is valid for R_i when provided $\overline{\upsilon}$ and $\mathbf{k}_L$.
With this, we compute $I(\Upsilon';\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L)$ as below:

$$\begin{aligned}
I(\Upsilon';\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L) &= E_{P(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)}\frac{P(\Upsilon',\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L)}{P(\Upsilon'|\overline{\Upsilon}^m,\mathcal{K}_L)P(\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L)}\\
&= \sum_{\upsilon'\in\Upsilon',\overline{\upsilon}\in\overline{\Upsilon}^m,\Lambda,\mathbf{k}_i\in\mathcal{K}_i,\mathbf{k}_L\in\mathcal{K}_L} P(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)\log\frac{P(\upsilon',\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)}{P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)}\\
&= \sum_{\substack{\upsilon'\in\Upsilon',\overline{\upsilon}\in\overline{\Upsilon}^m,\Lambda\\ \mathbf{k}_i\in\mathcal{K}_i,\mathbf{k}_L\in\mathcal{K}_L}} P(\upsilon',\overline{\upsilon},\mathbf{k}_L)P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\log\frac{P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)}{P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)}\\
&= \sum_{\upsilon'\in\Upsilon',\overline{\upsilon}\in\overline{\Upsilon}^m,\Lambda,\mathbf{k}_L\in\mathcal{K}_L,P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\neq 0} P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\\
&\times\log\frac{P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)}{P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)P(\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)}.
\end{aligned}$$

We note here that, when $P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\neq 0$, if $\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)=0$, then $P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)=0$. So, the summation taken over $\mathcal{K}_i$ above is restricted to all $\mathbf{k}_i$ for which $\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)=1$. Henceforth, we have:

$$I(\Upsilon';\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L) = \sum_{\substack{\upsilon'\in\Upsilon',\overline{\upsilon}\in\overline{\Upsilon}^m,\Lambda,\\ \mathbf{k}_L\in\mathcal{K}_L,P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\neq 0}} P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)$$
$$\times \log\frac{P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)}{P(\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)}$$
$$\geq \sum_{\substack{\upsilon'\in\Upsilon',\overline{\upsilon}\in\overline{\Upsilon}^m,\Lambda,\\ \mathbf{k}_L\in\mathcal{K}_L,P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\neq 0}} P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)$$
$$\times \log\frac{\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)}{\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)} \tag{16}$$

Again, we observe that, if $P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\neq 0$ and $\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)=0$, then $P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)=0$. This implies

$$\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\upsilon',\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L)=1 \tag{17}$$

and

$$\sum_{\mathbf{k}_i\in\mathcal{K}_i} P(\mathbf{k}_i|\overline{\upsilon},\mathbf{k}_L)\mathcal{X}_S(\upsilon',\overline{\upsilon},\mathbf{k}_i,\mathbf{k}_L) = P(\upsilon \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L). \tag{18}$$

Based on (17) and (18) above, we continue (16) and have:

$$I(\Upsilon';\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L) \geq -\sum_{\substack{\upsilon'\in\upsilon'\overline{\upsilon}\in\overline{\Upsilon}^m\\ \Lambda,\mathbf{k}_L\in\mathcal{K}_L}} P(\upsilon',\overline{\upsilon},\mathbf{k}_L)\log P(\upsilon \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L)$$
$$= -\sum_{\substack{\overline{\upsilon}\in\overline{\Upsilon}^m\\ \mathbf{k}_L\in\mathcal{K}_L}} P(\overline{\upsilon},\mathbf{k}_L)\big[\sum_{\substack{\upsilon'\in\Upsilon'\\ \Lambda}} P(\upsilon'|\overline{\upsilon},\mathbf{k}_L)\log P(\upsilon \text{ is valid for } R_i|\overline{\upsilon},\mathbf{k}_L)\big]. \tag{19}$$

Combining (19) and (15), we have $P_S[i,L]\geq 2^{-I(\Upsilon';\mathcal{K}_i|\overline{\Upsilon}^m,\mathcal{K}_L)}$ and therefore finish the proof of (iii) in the theorem. □

B Proof of Theorem 3

Firstly, from the construction description, it is easy to see that $\overline{\mathcal{K}}=M^\mu, \mathcal{K}_i=M, 1\leq i\leq N$ and there are μ tags required for each message. Therefore it suffices to prove $\overline{P_I}=P_I, \overline{P_{I_{sub}}}=P_{I_{sub}}$, and $\overline{P_S}=P_S$. We later use $\mathcal{K}^{\mu-1}$ to denote the collection of all $(\mu-1)$-tuple elements, each of which is from $\mathcal{K}$.
(i) For $\overline{P_I}$, we have:

$$\overline{P_I} = \max_{\mathbf{0}\neq\mathbf{v}\in\mathcal{S},\overline{\mathbf{t}}_{\mathbf{v}i}} \frac{|\{\overline{\mathbf{k}}\in\overline{\mathcal{K}^\mu} \mid \sum_{i=0}^{\mu-1} x_\mu^i\overline{\mathbf{t}}_{\mathbf{v}i}=\mathbf{v}P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|} = \max_{\mathbf{0}\neq\mathbf{v}\in\mathcal{S}} \frac{|\{\overline{\mathbf{k}}\in\overline{\mathcal{K}^\mu} \mid \sum_{i=0}^{\mu-1} x_\mu^i\mathbf{v}\overline{\mathbf{k}}_i=\mathbf{v}P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|}.$$
$$= \max_{\mathbf{0}\neq\mathbf{v}\in\mathcal{S}} \frac{|\{\overline{\mathbf{k}}\in\overline{\mathcal{K}^\mu} \mid \mathbf{v}\sum_{i=0}^{\mu-1} x_\mu^i\overline{\mathbf{k}}_i=\mathbf{v}P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|}.$$

We are going to prove the following equation:

$$\max_{\mathbf{0}\neq\mathbf{v}\in\mathcal{S}} \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^{\mu}} \mid \mathbf{v}\sum_{i=0}^{\mu-1} x_{\mu}^{i}\overline{\mathbf{k}}_i = \mathbf{v}P(\mu)\}|}{|\overline{\mathcal{K}^{\mu}}|} = \max_{\mathbf{0}\neq\mathbf{v}\in\mathcal{S}} \frac{|\{\mathbf{k}\in\mathcal{K} \mid \mathbf{v}\in\ker\psi_{\mathbf{k}}\}|}{|\mathcal{K}|}. \tag{20}$$

which implies that $\overline{P_I} = P_I$. We prove (20) as below.

– On one hand, for the denominator part of LHS in (20), we have:

$$|\overline{\mathcal{K}^{\mu}}| = |\{\overline{\mathbf{k}} \in \overline{\mathcal{K}}| \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j), 1 \le j \le \mu-1\}|$$

$$= |\cup_{\overline{\mathbf{k}}_0\in\mathcal{K}} \{(\overline{\mathbf{k}}_1,\ldots,\overline{\mathbf{k}}_{\mu-1}) \in \mathcal{K}^{\mu-1}| \sum_{i=1}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j) - \overline{\mathbf{k}}_0, 1 \le j \le \mu-1\}|. \tag{21}$$

In terms of (21), we have a claim as below:

Claim. For any fixed $\overline{\mathbf{k}}_0 \in \mathcal{K}$, there is a unique solution to $(\overline{\mathbf{k}}_1,\ldots,\overline{\mathbf{k}}_{\mu-1})$ such that

$$\sum_{i=1}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j) - \overline{\mathbf{k}}_0, 1 \le j \le \mu-1. \tag{22}$$

To see why, we can rewrite (22) into a system of linear equations shown in (23) below, where $\overline{\mathbf{k}}_1,\ldots,\overline{\mathbf{k}}_{\mu-1}$ are unknowns (recall that x_j is nonzero for all $1 \le j \le \mu-1$):

$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^{\mu-2} \\ 1 & x_2 & \cdots & x_2^{\mu-2} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{\mu-1} & \cdots & x_{\mu-1}^{\mu-2} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{k}}_1 \\ \overline{\mathbf{k}}_2 \\ \vdots \\ \overline{\mathbf{k}}_{\mu-1} \end{bmatrix} = \begin{bmatrix} x_1^{-1}\ (P(1) - \overline{\mathbf{k}}_0) \\ x_2^{-1}\ (P(2) - \overline{\mathbf{k}}_0) \\ \vdots \\ x_{\mu-1}^{-1}\ (P(\mu-1) - \overline{\mathbf{k}}_0) \end{bmatrix} \tag{23}$$

Now, since x_i's are distinct from each other, it is clear that there is a unique solution to $(\overline{\mathbf{k}}_1,\ldots,\overline{\mathbf{k}}_{\mu-1})$. We denote the unique solution as $(\overline{\mathbf{k}'}_1,\ldots,\overline{\mathbf{k}'}_{\mu-1})$.
With Claim B, we continue (21) and finally compute the denominator part of LHS in (20) as:

$$|\overline{\mathcal{K}^{\mu}}| = |\{\overline{\mathbf{k}} \in \overline{\mathcal{K}}| \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j), 1 \le j \le \mu-1\}| = |\cup_{\overline{\mathbf{k}}_0\in\mathcal{K}}| = |\mathcal{K}|. \tag{24}$$

– On the other hand, for the numerator part of LHS in (20), $\forall\ \mathbf{0} \neq \mathbf{v} \in \mathcal{S}$ we have:

$$|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^{\mu}}|\mathbf{v} \sum_{i=0}^{\mu-1} x_{\mu}^{i} \overline{\mathbf{k}}_i = \mathbf{v}P(\mu)\}|$$

$$= |\{\overline{\mathbf{k}} \in \overline{\mathcal{K}}| \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j), 1 \le j \le \mu-1, \mathbf{v} \sum_{i=0}^{\mu-1} x_{\mu}^{i} \overline{\mathbf{k}}_i = \mathbf{v}P(\mu)\}|.$$

As a further step, when we exploit Claim B and the aforementioned unique solution $(\overline{\mathbf{k}'}_1, \ldots, \overline{\mathbf{k}'}_{\mu-1})$, we have:

$$|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}} | \sum_{i=0}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j), 1 \leq j \leq \mu-1, \mathbf{v} \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{k}}_i = \mathbf{v}P(\mu)\}|$$

$$= |\cup_{\overline{\mathbf{k}}_0 \in \mathcal{K}} \mid \mathbf{v}\overline{\mathbf{k}}_0 = \mathbf{v}P(\mu) - \mathbf{v} \sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i \mid \leq |\cup_{\overline{\mathbf{k}}_0 \in \mathcal{K}} \mid \mathbf{v}\overline{\mathbf{k}}_0 = \mathbf{0} \mid = |\{\mathbf{k} \in \mathcal{K} \mid \mathbf{v} \in \ker \psi_{\mathbf{k}}\}|.$$

The inequality above holds thanks to $\mathcal{K}$ being a subspace over $\mathbb{F}_q$ and satisfying Property 1. Proof details are omitted due to space constraints and can be checked from the proof of Lemma 1 in [13] (or Lemma 3.1 in [15]).
Moreover, the "=" in the inequality above can always happen, sine it can happen that $\mathbf{v}P(\mu) - \mathbf{v} \sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i = \mathbf{0}$.
Finally, we compute numerator part of LHS in (20) as:

$$\max_{\mathbf{0} \neq \mathbf{v} \in \mathcal{S}} |\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v} \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{k}}_i = \mathbf{v}P(\mu)\}| = \max_{\mathbf{0} \neq \mathbf{v} \in \mathcal{S}} |\{\mathbf{k} \in \mathcal{K} \mid \mathbf{v} \in \ker \psi_{\mathbf{k}}\}|. \tag{25}$$

Combining (24) and (25), we prove (20)and thus prove that $\overline{P_I} = P_I$.

(ii) For $\overline{P_{I_{sub}}} = \max_{V, \overline{\mathbf{t}}_{\mathbf{v}_j i}} \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{t}}_{\mathbf{v}_j i} = \mathbf{v}_j P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|} = \max_V \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{k}_i} = \mathbf{v}_j P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|}$,
in a quite similar way as above for proving (20), we can prove the fact below:

$$\max_V \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{k}}_i = P(\mu)\}|}{|\overline{\mathcal{K}^\mu}|} = \max_V \frac{|\{\mathbf{k} \in \mathcal{K} \mid \mathbf{v}_j \in \ker \psi_{\mathbf{k}}\}|}{|\mathcal{K}|},$$

which implies that $\overline{P_{I_{sub}}} = P_{I_{sub}}$. Details are omitted due to tight space constraints.

(iii) We prove the theorem for $\overline{P_S}$ with $\overline{P_S} = \max_{V, \mathbf{v} \notin V, \overline{\mathbf{t}}_{\mathbf{v}_j i}, \overline{\mathbf{t}}_{\mathbf{v} i}} \frac{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j \overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}, \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{t}}_{\mathbf{v} i} = \mathbf{v}P(\mu)\}|}{|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j \overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}\}|}$.
Firstly, for a fixed $\mathbf{k}_0 \in \mathcal{K}$, if we write $A = \{(\overline{\mathbf{k}}_1, \ldots, \overline{\mathbf{k}}_{\mu-1}) \in \mathcal{K}^{\mu-1} | \sum_{i=1}^{\mu-1} x_j^i \overline{\mathbf{k}}_i = P(j) - \overline{\mathbf{k}}_0, 1 \leq j \leq \mu-1\}$, and $B = \{(\overline{\mathbf{k}}_1, \ldots, \overline{\mathbf{k}}_{\mu-1}) \in \mathcal{K}^{\mu-1} | \mathbf{v}_j \overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}, 1 \leq j \leq \mu-1\}$, then based on (21), we can rewrite the denominator part of $\overline{P_S}$ as:

$$|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j \overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}\}| = |\cup_{\overline{\mathbf{k}}_0 \in \mathcal{K}} \mid \mathbf{v}_j \overline{\mathbf{k}}_0 = \overline{\mathbf{t}}_{\mathbf{v}_j 0} \ (A \cap B)|. \tag{26}$$

Now, from Claim B above, we have $|A| = 1$ and more precisely, $A = \{(\overline{\mathbf{k}'}_1, \ldots, \overline{\mathbf{k}'}_{\mu-1})\}$ as mentioned above. So, if $(\overline{\mathbf{k}'}_1, \ldots, \overline{\mathbf{k}'}_{\mu-1}) \notin B$ then we demonstrate that the tagged massages $\{\mathbf{v}_j \overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}\}$, which are gathered by the corrupted nodes, are not useful; we ignore this case for computing P_S (due to the definition of P_S). Otherwise, $(\overline{\mathbf{k}'}_1, \ldots, \overline{\mathbf{k}'}_{\mu-1}) \in B$; in this case, we have $A \cap B = \{(\overline{\mathbf{k}'}_1, \ldots, \overline{\mathbf{k}'}_{\mu-1})\}$. With this, we continue (26) to compute denominator part of $\overline{P_S}$ as:

$$|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j \overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}\}| = |\cup_{\overline{\mathbf{k}}_0 \in \mathcal{K}} \mid \mathbf{v}_j \overline{\mathbf{k}}_0 = \overline{\mathbf{t}}_{\mathbf{v}_j 0} \mid = |\mathbf{k} \in \mathcal{K} \mid \mathbf{v}_j \mathbf{k} = \overline{\mathbf{t}}_{\mathbf{v}_j 0} \mid. \tag{27}$$

Furthermore, with all the above, we can rewrite the numerator part of $\overline{P_S}$ in a following way:

$$|\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j\overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}, \sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{t}}_{\mathbf{v}i} = \mathbf{v}P(\mu)\}| = |\{\overline{\mathbf{k}} \in \overline{\mathcal{K}^\mu} \mid \mathbf{v}_j\overline{\mathbf{k}}_i = \overline{\mathbf{t}}_{\mathbf{v}_j i}, \mathbf{v}\sum_{i=0}^{\mu-1} x_\mu^i \overline{\mathbf{k}}_i = \mathbf{v}P(\mu)\}|$$

$$= |\cup_{\overline{\mathbf{k}}_0 \in \mathcal{K}} \mid \mathbf{v}_j\overline{\mathbf{k}}_0 = \overline{\mathbf{t}}_{\mathbf{v}_j 0}, \mathbf{v}\overline{\mathbf{k}_0} = \mathbf{v}P(\mu) - \mathbf{v}\sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i \ \{(\overline{\mathbf{k}'}_1, \ldots, \overline{\mathbf{k}'}_{\mu-1})\}|$$

$$= |\cup_{\overline{\mathbf{k}}_0 \in \mathcal{K}} \mid \mathbf{v}_j\overline{\mathbf{k}}_0 = \overline{\mathbf{t}}_{\mathbf{v}_j 0}, \mathbf{v}\overline{\mathbf{k}_0} = \mathbf{v}P(\mu) - \mathbf{v}\sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i \mid$$

$$= |\mathbf{k} \in \mathcal{K} \mid \mathbf{v}_j\mathbf{k} = \overline{\mathbf{t}}_{\mathbf{v}_j 0}, \mathbf{v}\mathbf{k} = \mathbf{v}P(\mu) - \mathbf{v}\sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i \mid. \tag{28}$$

Taking (27) and (28), we have an expression for $\overline{P_S}$ as:

$$\overline{P_S} = \max_{V, v\notin V, \overline{\mathbf{t}}_{\mathbf{v}_j 0}} \frac{|\mathbf{k} \in \mathcal{K} \mid \mathbf{v}_j\mathbf{k} = \overline{\mathbf{t}}_{\mathbf{v}_j 0}, \mathbf{v}\mathbf{k} = \mathbf{v}P(\mu) - \mathbf{v}\sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i \mid}{|\mathbf{k} \in \mathcal{K} \mid \mathbf{v}_j\mathbf{k} = \overline{\mathbf{t}}_{\mathbf{v}_j 0} \mid}.$$

Recall that $\mathcal{K}$ is a subspace over $\mathbb{F}_q$ and satisfies Property 1. Moreover, it can happen that $\overline{\mathbf{t}}_{\mathbf{v}_j 0} = \mathbf{0}$ and $\mathbf{v}P(\mu) - \mathbf{v}\sum_{i=1}^{\mu-1} x_\mu^i \overline{\mathbf{k}'}_i = \mathbf{0}$. Combing these facts, we have $\overline{P_S} = P_S$. We finally complete the proof of the theorem. □