Secure Network Function Computation for Linear Functions—Part I: Source Security

Xuan Guang, Member, IEEE, Yang Bai, Student Member, IEEE, and Raymond W. Yeung, Fellow, IEEE

Abstract—In this paper, we put forward secure network function computation over a directed acyclic network. In such a network, a sink node is required to compute with zero error a target function of which the inputs are generated as source messages at multiple source nodes, while a wiretapper, who can access any one but not more than one wiretap set in a given collection of wiretap sets, is not allowed to obtain any information about a security function of the source messages. The secure computing capacity for the above model is defined as the maximum average number of times that the target function can be securely computed with zero error at the sink node with the given collection of wiretap sets and security function for one use of the network. The characterization of this capacity is in general overwhelmingly difficult. In the current paper, we consider securely computing linear functions with a wiretapper who can eavesdrop any subset of edges up to a certain size \( r \), referred to as the security level, with the security function being the identity function. We first prove an upper bound on the secure computing capacity, which is applicable to arbitrary network topologies and arbitrary security levels. This upper bound depends on the network topology and security level. Furthermore, we obtain an upper bound and a lower bound on this bound, which are both in closed form. In particular, when the security level \( r \) is equal to 0, our upper bound reduces to the computing capacity without security consideration. Also, we discover the surprising fact that for some models, there is no penalty on the secure computing capacity compared with the computing capacity without security consideration. Furthermore, we obtain an equivalent expression of the upper bound by using a graph-theoretic approach, and accordingly we develop an efficient approach for computing this bound. On the other hand, we present a construction of linear function-computing secure network codes and obtain a lower bound on the secure computing capacity. By our code construction, for the linear function which is over a given finite field, we can always construct a (vector-) linear function-computing secure network code over the same field. We also give some sufficient conditions for the tightness of the lower bound in terms of the network topology. With this lower bound and the upper bound we have obtained, the secure computing capacity for some classes of secure models can be fully characterized. Another interesting case is that the security function is the same as the target function, which will be investigated in Part II of this paper.

Index Terms—Secure network function computation, linear functions, source security, secure computing capacity, linear function-computing secure network coding, code construction, information-theoretic security.

I. INTRODUCTION

IN THIS paper, we investigate secure network function computation that incorporates information-theoretic security with zero-error network function computation. A general setup of the model with the single destination is presented as follows. In a directed acyclic graph \( G \), i) the single sink node \( r \) is required to compute repeatedly (with zero error) a function \( f \), called the target function, whose arguments are source messages generated at a set of source nodes \( S \); and ii) a wiretapper, who can access any one but not more than one edge subset \( W \subseteq \mathcal{W} \), is not allowed to obtain any information about another function \( \zeta \), called the security function, whose arguments are also the source messages. Here, \( W \) and \( \mathcal{W} \) are referred to as the wiretap set and the collection of wiretap sets, respectively. The graph \( G \), together with \( S \) and \( \rho \), forms a network \( \mathcal{N} \), and we use the quadruple \( (\mathcal{N}, f, W, \zeta) \) to denote this model of secure network function computation.

We note that when both the target function \( f \) and the security function \( \zeta \) are the identity function, the secure model \( (\mathcal{N}, f, W, \zeta) \) degenerates to a secure network coding model (cf. [1], [2], [3], [4], [5], [6]). When no security constraint is considered, i.e., \( W = \emptyset \), the secure model \( (\mathcal{N}, f, W, \zeta) \) degenerates to the model of network function computation (cf. [7], [8], [9], [10], [11], [12], [13], [14]).

A. Related Works

Information-theoretic security dates from Shannon’s celebrated paper [15], in which the well-known Shannon cipher system was studied. In this system, a sender wishes to transmit a private message to a receiver via a “public” channel which is eavesdropped by a wiretapper, and it is required that
this wiretapper cannot obtain any information about the private message. For this purpose, the sender applies a random key to encrypt the message and then transmits this encrypted message via the “public” channel. The random key is shared with the receiver via a “secure” channel that is inaccessible by the wiretapper. The receiver can recover the private message from the encrypted message and the random key, while the wiretapper cannot obtain any information about the private message. Another well-known cipher system of information-theoretic security is secret sharing, proposed independently by Blakley [16] and Shamir [17]. In this system, a secret is encoded into shares which are distributed among a set of participants, and only the qualified subsets of participants can recover the secret, while no information at all about the secret can be obtained from the shares of any unqualified set of participants. The Shannon cipher system can be regarded as a special case of secret sharing. Wiretap channel II, proposed by Ozarow and Wyner [18], is a related system of information-theoretic security, in which the sender needs to transmit the private message to the receiver through a set of noiseless point-to-point channels without leaking any information about the private message to a wiretapper who can fully access any one but not more than one subset of the channels up to a certain size. Wiretap channel II can be regarded as a special case of secret sharing.

In the paradigm of network coding, the information-theoretically secure problem in the presence of a wiretapper, called secure network coding, was introduced by Cai and Yeung in [1] and [2]. In the wiretap network model of secure network coding, the source node multicasts the source message to all the sink nodes which as legal users are required to decode the source message with zero error; while the wiretapper, who can access any one wiretap set of edges, is not allowed to obtain any information about the source message. Logically, the foregoing three classical information-theoretically secure systems, the Shannon cipher system, secret sharing and wiretap channel II can be formulated as special cases of the wiretap network model of secure network coding.

Network function computation was considered in the literature [10], [11], [12], [13], [14]. In the model with a single destination, the sink node is required to repeatedly compute with zero error a target function of the source messages generated at multiple source nodes over a directed acyclic network. When the sink node is required to compute the identity function of the source messages, or equivalently, recover the source messages, the problem degenerates to the network coding problem [19], [20], [21], [22], [23]. Appuswamy et al. [10] investigated the fundamental computing capacity, i.e., the maximum average number of times that the function can be computed with zero error for one use of the network, and gave a cut-set bound that is valid under certain constraints on either the network topology or the target function. Huang et al. [13] obtained an enhancement of Appuswamy et al.’s bound that can be applied for arbitrary functions and arbitrary network topologies. For the special cases, e.g., computing an arbitrary function over a multi-edge tree network and computing the identity function or the algebraic sum function over an arbitrary network topology, the upper bounds in [10] and [13] coincide and are tight. Guang et al. [14] proved an improved upper bound on the computing capacity by using a novel approach based on cut-set strong partition. This improved upper bound is applicable to arbitrary target functions and arbitrary network topologies. In particular, the improved upper bound is asymptotically achievable for all previously considered network function computation problems whose computing capacities are known.

B. Contributions and Organization of the Paper

For the model of secure network function computation \((N, f, W, \zeta)\), we consider in this paper the special case that the security function \(\zeta\) is the identity function, namely that we need to protect the information sources from being leaked to the wiretapper. We use \((N, f, W)\) to denote this special case of the model. The notion of security is considered in almost all previously studied information-theoretic security models, e.g., the Shannon cipher system [15], the secret sharing [16], [17], the wiretap channel II [18], and secure network coding [2], [3]. Another interesting case is that the security function is the same as the target function, i.e., \(\zeta = f\). This will be treated in Part II of the current paper.

From the information theoretic point of view, we are interested in characterizing the secure computing capacity for \((N, f, W)\), which is defined as the maximum average number of times that the function \(f\) can be securely computed with zero error at the sink node with the given collection of wiretap sets \(W\) for one use of the network \(N\). However, characterizing this secure computing capacity for an arbitrary security model \((N, f, W)\) is overwhelmingly difficult and complicated, and even for the simpler case of network function computation without any security consideration, the capacity characterization is still open [14]. Thus, in the paper we focus on the model of securely computing a linear function over a finite field on an arbitrary network where the wiretapper can eavesdrop any one edge subset up to a size \(r\), referred to as the security level. We note that the linear functions are not only an important class of target functions but also the only class of non-identity functions whose computing capacities can be characterized without any security consideration.

In this paper, we first prove a non-trivial upper bound on the secure computing capacity, which is applicable to arbitrary network topologies and arbitrary security levels. This upper bound improves the upper bound we previously obtained in [24]. In particular, when no security is considered, i.e., \(r = 0\), our upper bound reduces to the capacity for computing a linear function over the network without security consideration, which we call the computing capacity for simplicity. Furthermore, we discover that for some models, there is no penalty on the secure computing capacity compared with the computing capacity without security consideration. In other words, the secure computing capacity may coincide with the computing capacity for a security level \(r > 0\). From the upper bound on the secure computing capacity we have obtained, we also obtain a non-trivial upper bound on the maximum security level such that the function can be securely computed with a positive rate.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Our upper bound on the secure computing capacity, which depends on the network topology and security level, is graph-theoretic and not in closed form. In order to compute this upper bound efficiently, we prove an equivalent expression by using a graph-theoretic approach. Accordingly, we devise an algorithm for computing this bound whose computational complexity is in a linear time of the number of edges in the network. We further prove an upper bound and a lower bound on this upper bound on the secure computing capacity, which are both in closed form.

Furthermore, we put forward a construction of linear (function-computing) secure network codes and thus obtain a lower bound on the secure computing capacity. By our code construction, with the target function being a linear function over a given finite field, we can always construct a (vector-)linear secure network code over the same field with rate up to the obtained lower bound on the secure computing capacity. Besides, we give two sufficient conditions for the tightness of the lower bound in terms of the network topology. With this lower bound and the upper bound we have obtained, we can fully characterize the secure computing capacity for some classes of secure models. Finally, an example is given that not only illustrates our code construction but also shows the existence of a linear secure network code that cannot be obtained by our code construction.

The paper is organized as follows. In Section II, we formally present the model of secure network function computation and function-computing secure network coding. The upper bound is proved in Section III, followed by an efficient graph-theoretic approach for computing the upper bound. Section IV is devoted to linear secure network coding for the model of secure network function computation, including the lower bound on the secure computing capacity, the code construction, the verification of the computability and security conditions, and the upper bounds on the minimum required field size for the code construction. In Section V, we conclude with a summary of our results and a remark on future research.

II. PRELIMINARIES

A. Model of Secure Network Function Computation

Let \( G = (V, E) \) be a directed acyclic graph, where \( V \) and \( E \) are a finite set of nodes and a finite set of edges, respectively. We allow multiple edges between two nodes and assume that a symbol taken from a finite alphabet \( B \) can be reliably transmitted on each edge for each use, i.e., we take the capacity of each edge to be 1 with respect to the alphabet \( B \). For an edge \( e \in E \), the tail node and head node of \( e \) are denoted by \( \text{tail}(e) \) and \( \text{head}(e) \), respectively. For a node \( u \in V \), we let \( \text{In}(u) = \{ e \in E : \text{head}(e) = u \} \) and \( \text{Out}(u) = \{ e \in E : \text{tail}(e) = u \} \).

In the graph \( G \), if a sequence of edges \( (e_1, e_2, \ldots, e_m) \) satisfies \( \text{tail}(e_i) = u, \text{head}(e_{i+1}) = v \) for \( i = 1, 2, \ldots, m \), then \( (e_1, e_2, \ldots, e_m) \) is called a path from the node \( u \) (or the edge \( e_1 \)) to the node \( v \) (or the edge \( e_m \)). In particular, a single edge \( e \) is regarded as a path from \( \text{tail}(e) \) to \( \text{head}(e) \) (or from \( e \) to itself). Furthermore, we write \( d \rightarrow e (or \ tail(d) \rightarrow head(e), tail(d) \rightarrow e, d \rightarrow head(e)) \) if there exists a path from edge \( d \) to edge \( e \). We now consider two disjoint subsets of nodes \( U \) and \( V \). An edge subset \( C \subseteq E \) is called a cut separating \( V \) from \( U \) if for each pair \( (u, v) \) of \( u \in U \) and \( v \in V \), no path exists from \( u \) to \( v \) upon removing the edges in \( C \). In particular, if \( U = \{ u \} \) and \( V = \{ v \} \) are two disjoint singleton subsets of nodes, a cut separating \( V \) from \( U \) is called a cut separating node \( v \) from node \( u \). The capacity of a cut separating \( V \) from \( U \) is defined as the size of this cut. A cut \( C \) separating \( V \) from \( U \) is called a minimum cut separating \( V \) from \( U \) if there does not exist a cut \( C' \) separating \( V \) from \( U \) such that \( |C'| < |C| \). The capacity of a minimum cut separating \( V \) from \( U \), denoted by \( \text{mincut}(U, V) \).

We let \( S \subseteq V \) be the set of source nodes \( \sigma_1, \sigma_2, \ldots, \sigma_s \) and \( \rho \in V \setminus S \) be the single sink node, where each source node \( \sigma_i \) has no input edges and the sink node \( \rho \) has no output edges, i.e., \( \text{In}(\sigma_i) = \text{Out}(\rho) = \emptyset, i = 1, 2, \ldots, s \). We further assume that there exists a directed path from every node \( u \in V \setminus \{ \rho \} \) to \( \rho \) in \( G \). Then, the source nodes are all the nodes in \( V \) without input edges and the sink node \( \rho \) is the unique node in \( V \) without output edges. The graph \( G \), together with \( S \) and \( \rho \), forms a network \( N \), i.e., \( N = (G, S, \rho) \).

Let \( f : \prod_{i=1}^{m} A_i \rightarrow \mathcal{Q} \) be a nonconstant function, called the target function, that is needed to be computed with zero error at the sink node \( \rho \), where \( A_i, 1 \leq i \leq s \) and \( \mathcal{Q} \) are all finite alphabets. Further, we assume without loss of generality that \( (j \text{th argument of the target function } f \text{ is generated at the } i \text{th source node } \sigma_i, i = 1, 2, \ldots, s \). Let \( \ell \) and \( n \) be two positive integers. We consider computing the target function \( f \) times by using the network \( n \) times, i.e., by transmitting at most \( n \) symbols in \( B \) on each edge in \( E \). For each \( i = 1, 2, \ldots, s \), we let the information source at the \( i \)th source node \( \sigma_i \) be a random variable \( M_i \) according to the uniform distribution on \( A_i \). All the sources \( M_i, 1 \leq i \leq s \) are mutually independent. Let \( M_S = (M_1, M_2, \ldots, M_s) \) be the source message vector generated by \( S \). The \( \ell \) values of the target function \( f \)

\[
\mathbf{f}(M_S) = \left( f(M_{1,j}, M_{2,j}, \ldots, M_{s,j}) : j = 1, 2, \ldots, \ell \right)
\]  

are required to be computed at \( \rho \) with zero error.

In addition, we consider a collection of edge subsets \( W \) where each edge subset \( W \subseteq W \) is called a wiretap set, and another nonconstant function \( \zeta : \prod_{i=1}^{m} A_i \rightarrow \mathcal{Q} \), called the security function, where \( \mathcal{Q} \) is a finite alphabet. In this model, when the \( \ell \) target function values \( \mathbf{f}(M_S) \) are computed at \( \rho \) through the network \( N \), the \( \ell \) values of the security function \( \zeta(M_S) = \left( \zeta(M_{1,j}, M_{2,j}, \ldots, M_{s,j}) : j = 1, 2, \ldots, \ell \right) \) are required to be protected from a wiretapper who can access any one but not more than one wiretap set \( W \subseteq W \). The collection of wiretap sets \( W \) and the security function \( \zeta \) are known
by the source nodes and the sink node but which wiretap set in $W$ is eavesdropped by the wiretapper is unknown. We have completed the specification of our model of secure network function computation, denoted by $(N, f, W, \zeta)$.

In this paper, we consider a special case of our model that the security function $\zeta$ is the identity function, namely that we need to protect the information sources from being leaked to the wiretapper. We use $(N, f, W)$ to denote this special case of the model. The notion of security is considered in almost all previously studied information-theoretic security models, e.g., the Shannon cipher system [15], the secret sharing [16], [17], the wiretap channel II [18], and secure network coding [2].

In the next subsection, we will define a function-computing secure network code on $(N, f, W)$.

Another interesting case of the model of secure network function computation is that the security function is the same as the target function, i.e., $\zeta = f$. This case will be treated in the next paper.

B. Function-Computing Secure Network Code

In order to combat the wiretapper in the secure model $(N, f, W)$, it is necessary to randomize the source messages, which is similar to the previously studied information-theoretic security models (e.g., the Shannon cipher system [15], the secret sharing [16], [17], the wiretap channel II [18], and secure network coding [2]). The reason is as follows. Suppose no randomness is used to randomize the source messages. Then, the message transmitted on each edge is a function of the source message vector $M_S$ and hence is not independent of $M_S$ unless the transmitted message takes a constant value, which is equivalent to transmitting nothing on that edge. If this is the case, nothing can be computed at the sink node. Therefore, randomness is necessary for transmitting a message securely.

As part of the network code to be defined, we assume that for $i = 1, 2, \cdots, s$, a random variable $K_i$, called a key, which is distributed uniformly on a finite set $K_i$, is available to the source node $\sigma_i$. We let $K_S = (K_1, K_2, \cdots, K_s)$. Further, assume that all the keys $K_i$ and the source messages $M_i$, $i = 1, 2, \cdots, s$, are mutually independent.

Now, we consider securely computing the target function $f$ times by using the network $n$ times under the collection of wiretap sets $W$. An $(\ell, n)$ function-computing secure network code for $(N, f, W)$ is defined as follows. First, we let $m_i \in A_i^\ell$ and $k_i \in K_i$ be arbitrary outputs of the source message $M_i$ and the key $K_i$, respectively, for $i = 1, 2, \cdots, s$. Accordingly, let $M_S = (m_1, m_2, \cdots, m_s)$ and $K_S = (k_1, k_2, \cdots, k_s)$, which can be regarded as two arbitrary outputs of $M_S$ and $K_S$, respectively. An $(\ell, n)$ secure network code $C$ consists of a local encoding function $\theta_e$ for each edge $e \in \mathcal{E}$, where

$$
\theta_e : \bigcup_{d \in \mathcal{D}_{\mathcal{L}(t)}} A_i^\ell \times K_i \rightarrow B^n, \quad \text{if } \text{tail}(e) = \sigma_i \text{ for some } i,
$$

and a decoding function $\varphi : \prod_{d \in \mathcal{D}_{\mathcal{L}(t)}} B^n \rightarrow \mathcal{O}^\ell$ at the sink node $\rho$, which is used to compute the target function $f$ with zero error. Furthermore, let $y_e \in B^n$ be the message transmitted on each edge $e \in \mathcal{E}$ by using the code $C$ under the source message vector $M_S$ and the key vector $K_S$. With the encoding mechanism as described in (2), we readily see that $y_e$ is a function of $M_S$ and $K_S$, denoted by $\hat{g}_e(M_S, K_S)$ (i.e., $y_e = \hat{g}_e(M_S, K_S)$), where $\hat{g}_e$ can be obtained by recursively applying the local encoding functions $\theta_e$, $e \in \mathcal{E}$. More precisely, for each $e \in \mathcal{E}$,

$$
\hat{g}_e(M_S, K_S) = \begin{cases} 
\hat{g}_e(m_i, k_i), & \text{if } \text{tail}(e) = \sigma_i \text{ for some } i, \\
\hat{g}_e(\hat{g}_{\mathcal{D}_{\mathcal{L}(t)}}(M_S, K_S)), & \text{otherwise},
\end{cases}
$$

where $u = \text{tail}(e)$ and $\hat{g}_e(M_S, K_S) = \hat{g}_e(M_S, K_S) : e \in \mathcal{E}$ for an edge subset $E \subseteq \mathcal{E}$ (in particular, $\hat{g}_{\mathcal{D}_{\mathcal{L}(t)}}(M_S, K_S) = \hat{g}_e(M_S, K_S) : e \in \mathcal{D}_{\mathcal{L}(t)}$). We call $\hat{g}_e$ the global encoding function of the edge $e$ for the code $C$.

For the secure model $(N, f, W)$, we say an $(\ell, n)$ secure network code $C = \{\theta_e : e \in \mathcal{E} ; \varphi \}$ is admissible if the following computability and security conditions are satisfied:

• **computability condition**: the sink node $\rho$ computes the target function $f$ with zero error, i.e., for all $M_S \in \prod_{i=1}^{s} A_i^\ell$ and $K_S \in \prod_{i=1}^{s} K_i$,

$$
\varphi(\hat{g}_{\mathcal{D}_{\mathcal{L}(t)}}(M_S, K_S)) = f(M_S);
$$

• **security condition**: for any wiretap set $W \in \mathcal{W}$, $Y_W$ and $M_S$ are independent, i.e.,

$$
I(Y_W; M_S) = 0,
$$

where $Y_W = (Y_e : e \in W)$, and $Y_e \triangleq \hat{g}_e(M_S, K_S)$ is the random variable transmitted on the edge $e$.

The secure computing rate of such an admissible $(\ell, n)$ secure network code $C$ is defined by

$$
R(C) \triangleq \frac{\ell}{n},
$$

i.e., the average number of times the function $f$ can be securely computed with zero error at $\rho$ under the collection of wiretap sets $W$ for one use of the network $N$. Further, we say that a nonnegative real number $R$ is achievable if $\forall \epsilon > 0$, there exists an admissible $(\ell, n)$ secure network code $C$ such that

$$
R(C) = \frac{\ell}{n} > R - \epsilon.
$$

We readily see that the secure computing rate $\ell/n$ of an admissible $(\ell, n)$ secure network code must be achievable. The secure computing rate region for the secure model $(N, f, W)$ is defined as

$$
\mathcal{R}(N, f, W) \triangleq \left\{ R : R \text{ is achievable for } (N, f, W) \right\},
$$

which is evidently closed and bounded. Consequently, the secure computing capacity for $(N, f, W)$ is defined as

$$
\bar{C}(N, f, W) \triangleq \max \mathcal{R}(N, f, W).
$$
C. Secure Network Function Computation for Linear Functions With Source Security

From the information theoretic point of view, we are interested in the secure computing capacity \( \hat{C}(\mathcal{N}, f, W) \) for an arbitrary model \( (\mathcal{N}, f, W) \). However, characterizing this capacity with the general setup is overwhelmingly difficult and complicated, and even for the simpler case of network function computation without any security consideration, the computing capacity characterization is still open (cf. [14]). In fact, for computing an arbitrary target function over an arbitrary network without any security consideration, characterizing the computing capacity is in general difficult. So far, only the network coding problem. As secure network function computing the identity function degenerates to the regular computing capacity characterization is still open (cf. [14]). In fact, computing the identity function degenerates to the regular network coding problem. As secure network function computation is a generalization of network function computation, we naturally consider secure network computing for the target functions whose computing capacities have been determined when no security is considered. Thus, in this paper we focus on linear functions over finite fields and for identity functions have been fully characterized. Note that computing the identity function degenerates to the regular network coding problem. As secure network function computation is a generalization of network function computation, we naturally consider secure network computing for the target functions whose computing capacities have been determined when no security is considered. Thus, in this paper we focus on linear functions over finite fields. This is not only an important class of target functions but also the only non-trivial class of functions whose computing capacities without any security consideration can be determined. To be specific, the target function \( f \) is a linear function over a finite field \( \mathbb{F}_q \) of form

\[
f(m_1, m_2, \cdots, m_s) = \sum_{i=1}^{s} a_i \cdot m_i,
\]

where \( m_i, a_i \in \mathbb{F}_q \) for all \( i = 1, 2, \cdots, s \) (i.e., \( A_i = \mathcal{O} = \mathbb{F}_q \), \( \forall \ 1 \leq i \leq s \) and \( f : \mathbb{F}_q^s \rightarrow \mathbb{F}_q \)). Accordingly, all the independent information sources \( M_i, i = 1, 2, \cdots, s \) are random variables distributed uniformly on the finite field \( \mathbb{F}_q \). Moreover, we consider a special collection of wiretap sets

\[
\mathcal{W}_r = \{ W \subseteq \mathcal{E} : 0 \leq |W| \leq r \},
\]

i.e., the wiretapper can eavesdrop any one subset of edges in the network up to a size \( r \), which is referred to as the security level. We remark that the empty set \( \emptyset \), regarded as the wiretap set of size 0, is in \( \mathcal{W}_r \). For notational simplicity, we write the secure model \( (\mathcal{N}, f, \mathcal{W}_r) \) and secure computing capacity \( \hat{C}(\mathcal{N}, f, \mathcal{W}_r) \) as \( (\mathcal{N}, f, r) \) and \( \hat{C}(\mathcal{N}, f, r) \), respectively. In the rest of the paper, we assume without loss of generality that \( B = \mathbb{F}_q \), i.e., an element in the field \( \mathbb{F}_q \) can be reliably transmitted on each edge for each use.

III. UPPER BOUND ON THE SECURE COMPUTING CAPACITY

We first observe that the characterization of secure computing capacity for a linear function over a finite field is equivalent to the characterization of secure computing capacity for an algebraic sum over the same field. This is explained as follows. We consider the model of secure network function computation \( (\mathcal{N}, f, r) \) with \( f \) being a linear function over a finite field \( \mathbb{F}_q \), i.e.,

\[
f(m_1, m_2, \cdots, m_s) = \sum_{i=1}^{s} a_i \cdot m_i.
\]

For the indices \( i \) with \( a_i = 0 \), we remove the terms \( a_i \cdot m_i \) from \( f \) to form a new linear function \( f' \), and at the same time remove the source nodes \( \sigma_i \) together with the output edges in \( \text{Out}(\sigma_i) \) from the graph \( \mathcal{G} \) to form a new graph \( \mathcal{G}' \). We update the network \( \mathcal{N}' \) to a new one

\[
\mathcal{N}' = (\mathcal{G}', \mathcal{S}' \setminus \{ \sigma_i : a_i = 0 \}, \rho).
\]

We readily see that with the security level \( r \), securely computing \( f \) over \( \mathcal{N}' \) is equivalent to securely computing \( f' \) over \( \mathcal{N}' \), and \( \hat{C}(\mathcal{N}, f, r) = \hat{C}(\mathcal{N}', f', r) \). Thus, we can assume without loss of generality that in the model \( (\mathcal{N}, f, r) \) with \( f(m_1, m_2, \cdots, m_s) = \sum_{i=1}^{s} a_i \cdot m_i \), we have \( a_i \neq 0 \) for all \( 1 \leq i \leq s \). Now, let \( x_i = a_i \cdot m_i \) and consider the algebraic sum \( g(x_1, x_2, \cdots, x_s) = \sum_{i=1}^{s} x_i \) over \( \mathbb{F}_q \). Since \( a_i \neq 0 \) for all \( 1 \leq i \leq s, m_i \) and \( x_i \) can be determined from each other, and hence an admissible \((f, n)\) secure network code for \((\mathcal{N}, f, r)\) can be readily modified into an admissible \((f, n)\) secure network code for \((\mathcal{N}, g, r)\), and vice versa. So, we have \( \hat{C}(\mathcal{N}, f, r) = \hat{C}(\mathcal{N}, g, r) \). By the above discussion, we can see that in order to investigate the secure network function computation for a linear function, it suffices to consider secure network function computation for an algebraic sum. Accordingly, in the rest of the paper, we let the target function \( f \) be the algebraic sum over a finite field \( \mathbb{F}_q \), i.e.,

\[
f(m_1, m_2, \cdots, m_s) = \sum_{i=1}^{s} m_i. \tag{5}
\]

A. The Upper Bound

Next, we present some graph-theoretic notations. Given a set of edges \( C \subseteq \mathcal{E} \), we define three subsets of the source nodes as follows:

\[
\mathcal{D}_C = \{ \sigma \in \mathcal{S} : \exists e \in C \text{ s.t. } \sigma \rightarrow e \},
\]

\[
\mathcal{I}_C = \{ \sigma \in \mathcal{S} : \sigma \rightarrow \rho \text{ upon deleting the edges in } C \text{ from } \mathcal{E} \},
\]

\[
\mathcal{J}_C = \mathcal{D}_C \setminus \mathcal{I}_C,
\]

where \( \sigma \rightarrow \rho \) denotes that there exists no path from \( \sigma \) to \( \rho \). Since we assume that there is a directed path from every node \( u \in \mathcal{V} \setminus \{ \rho \} \) to \( \rho \), in particular, a directed path from every source node \( \sigma \in \mathcal{S} \) to \( \rho \), we see that \( \mathcal{I}_C \subseteq \mathcal{D}_C \). Here, \( \mathcal{J}_C \) is the subset of the source nodes \( \sigma \) satisfying that there exists not only a path from \( \sigma \) to \( \rho \) passing through an edge in \( C \) but also a path from \( \sigma \) to \( \rho \) not passing through any edge in \( C \).\footnote{Similarly, since it is assumed that there is a directed path from every node \( u \in \mathcal{V} \setminus \{ \rho \} \) to \( \rho \), there is a path from \( \sigma \) to \( \rho \) passing through an edge \( e \in C \) provided that there is a path from \( \sigma \) to \( e \).}

In the network \( \mathcal{N} \), an edge set \( C \) is said to be a cut set if \( \mathcal{I}_C \neq \emptyset \), and we let \( \Lambda(\mathcal{N}) \) be the family of all the cut sets, i.e.,

\[
\Lambda(\mathcal{N}) = \{ C \subseteq \mathcal{E} : \mathcal{I}_C \neq \emptyset \}.
\]

In particular, we say a cut set \( C \) is a global cut set if \( \mathcal{I}_C = \mathcal{S} \). As an illustration, we consider the network in Fig. 1 to be discussed in Example 1. We let \( C = \{ e_1 \} \) be an edge subset. Then it is easily seen that \( \mathcal{D}_C = \{ \sigma_1, \sigma_2 \} \), \( \mathcal{I}_C = \{ \sigma_1 \} \) and \( \mathcal{J}_C = \{ \sigma_2 \} \). For another edge subset \( C' = \{ e_1, e_2 \} \), we readily see that \( \mathcal{I}_C = \{ \sigma_1, \sigma_2 \} = \mathcal{S} \), and so \( C' \) is a global cut set.

\[\text{Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.}\]
Now, we present an upper bound on the secure computing capacity \( \hat{C}(N, f, r) \) in the following theorem.

**Theorem 1:** Consider a model of secure network function computation \((N, f, r)\), where the target function \( f \) is an algebraic sum over a finite field \( \mathbb{F}_q \). Then,

\[
\hat{C}(N, f, r) \leq \min_{(W,C)\subseteq V \times X_{\mathcal{M}(N)}} \left( |C| - |W| \right). \tag{6}
\]

**Proof:** We let \( \hat{C} \) be an arbitrary admissible \((\ell, n)\) secure network code for the secure model \((N, f, r)\), of which all the global encoding functions are \( \tilde{g}_e, e \in \mathcal{E} \) (cf. (3)). We consider an arbitrary cut set \( C \in \Lambda(N) \), i.e., \( I_C \neq \emptyset \). Recall that all the source messages \( M_{i,j} \) for \( i = 1, 2, \ldots, s \) and \( j = 1, 2, \ldots, \ell \) are i.i.d. according to the uniform distribution on \( \mathbb{F}_q \). Thus, we have

\[
H \left( \sum_{i \in I_C} M_i \right) = \ell \cdot H \left( \sum_{i \in I_C} M_i \right) = \ell \cdot \log q, \tag{7}
\]

where we recall that \( M_i \) is the vector of the \( \ell \) random variables \( M_{i,j}, 1 \leq j \leq \ell \) with generic random variable \( M_i = (M_{i,1}, M_{i,2}, \ldots, M_{i,\ell}) \), and similar to (1), we let

\[
\sum_{i \in I_C} M_i \triangleq \left( \sum_{i \in I_C} M_{i,j} : j = 1, 2, \ldots, \ell \right).
\]

Now, we consider an arbitrary wiretap set \( W \subseteq C \) that satisfies \( 0 \leq |W| \leq r \) and \( D_W \subseteq I_C \). We remark that the empty set is also such a wiretap set. By the security condition (4), we have

\[
H(M_S) = H(M_S|Y_W),
\]

which implies that

\[
H(M_{I_C}) = H(M_{I_C}|Y_W), \tag{8}
\]

where we let \( M_{I_C} \triangleq (M_i : i \in I_C) \). It follows from (8) that

\[
H \left( \sum_{i \in I_C} M_i \right) = H \left( \sum_{i \in I_C} M_i \right| Y_W),
\]

or equivalently,

\[
H \left( \sum_{i \in I_C} M_i \right) + H(Y_W) = H \left( \sum_{i \in I_C} M_i, Y_W \right), \tag{9}
\]

because \( \sum_{i \in I_C} M_i \) is a function of \( M_{I_C} \).

By the definition of \( D_W \), there exists no path from any source node \( \sigma \in S \setminus D_W \) to any edge in \( W \). Then, we see that i) \( Y_W \) is independent of \( M_{S \setminus I_C} \) and \( K_{S \setminus I_C} \); ii) \( Y_W = \tilde{g}_W(M_S, K_S) \) depends only on \( (M_{D_W}, K_{D_W}) \). Following ii), we write

\[
Y_W = \tilde{g}_W(M_S, K_S) = \tilde{g}_W(M_{D_W}, K_{D_W}). \tag{10}
\]

Since \( D_W \subseteq I_C \), \( Y_W \) is also independent of \( M_{S \setminus I_C} \) and \( K_{S \setminus I_C} \). Hence, by (10) and the independence of \( (M_{I_C}, K_{I_C}) \) and \( (M_{S \setminus I_C}, K_{S \setminus I_C}) \), we obtain that \( \left( \sum_{i \in I_C} M_i, Y_W \right) \) and \( (M_{S \setminus I_C}, K_{S \setminus I_C}) \) are independent, which immediately implies that

\[
H \left( \sum_{i \in I_C} M_i, Y_W \right) = \sum_{i \in I_C} H(M_i, Y_W|\mathbb{M}_{S \setminus I_C}, K_{S \setminus I_C}), \tag{11}
\]

Continuing from (11), we consider

\[
H \left( \sum_{i \in I_C} M_i, Y_W |\mathbb{M}_{S \setminus I_C}, K_{S \setminus I_C} \right)
= H \left( \sum_{i \in I_C} M_i, Y_W, M_{S \setminus I_C}, K_{S \setminus I_C} \right)
+ H(Y_W |\mathbb{M}_{S \setminus I_C}, K_{S \setminus I_C})
= H \left( \sum_{i \in I_C} M_i, Y_W, M_{S \setminus I_C}, K_{S \setminus I_C} \right) + H(Y_W), \tag{12}
\]

where the equality (12) again follows from the independence of \( Y_W \) and \( (M_{S \setminus I_C}, K_{S \setminus I_C}) \). Now, we combine the equalities (9), (11) and (12) to obtain

\[
H \left( \sum_{i \in I_C} M_i \right) = H \left( \sum_{i \in I_C} M_i | Y_W, M_{S \setminus I_C}, K_{S \setminus I_C} \right). \tag{13}
\]

Next, we will prove the equality

\[
H \left( \sum_{i \in I_C} M_i | Y_C, M_{S \setminus I_C}, K_{S \setminus I_C} \right) = 0. \tag{14}
\]

We first consider the edge subset \( C' \triangleq \bigcup_{i \in S \setminus I_C} \text{Out}(\sigma_i) \). Then, \( D_{C'} = I_{C'} = S \setminus I_C \), So, we see that \( Y_{C'} \) depends only on \( M_{S \setminus I_C} \) and \( K_{S \setminus I_C} \), and similar to (10), we write

\[
Y_{C'} = \tilde{g}_{C'}(M_S, K_S) = \tilde{g}'_{C'}(M_{S \setminus I_C}, K_{S \setminus I_C}).
\]

By the above discussion, we have

\[
H \left( \sum_{i \in I_C} M_i | Y_C, M_{S \setminus I_C}, K_{S \setminus I_C} \right)
= \sum_{i \in I_C} H(M_i | Y_C, M_{S \setminus I_C}, K_{S \setminus I_C})
= \sum_{i \in I_C} H(M_i | Y_{C'}, M_{S \setminus I_C}, K_{S \setminus I_C}), \tag{15}
\]

where \( \hat{C} = C \cup C' \). Clearly, \( \hat{C} \) is a global cut set, namely that \( \hat{C} \) separates \( \tilde{g} \) from all the source nodes in \( S \). Together with the acyclicity of the graph \( \mathcal{G} \), we see that \( Y_{I_{\mathcal{G}}(\rho)} = \tilde{g}_{I_{\mathcal{G}}(\rho)}(M_S, K_S) \) is a function of \( Y_{\hat{C}} = \tilde{g}_{\hat{C}}(M_{S \setminus I_C}, K_{S \setminus I_C}) \) together with the admissibility of the secure network code \( \hat{C} \), the target function \( f \) must be computable with zero error on the global cut set \( \hat{C} \), i.e.,

\[
0 = H(f(M_S)|Y_{\hat{C}}) = H \left( \sum_{i \in S} M_i | Y_{\hat{C}} \right). \tag{16}
\]

because otherwise \( f \) cannot be computed at \( \rho \) with zero error. Now, we continue from (15) to obtain

\[
H \left( \sum_{i \in I_C} M_i | Y_C, M_{S \setminus I_C}, K_{S \setminus I_C} \right).
\]
\[= H \left( \sum_{i \in I_C} M_i \big| Y_{\bar{C}}, M_S \setminus I_C, K_{S \setminus I_C} \right) \]

\[= H \left( \sum_{i \in I_C} M_i \big| Y_{\bar{C}}, \sum_{i \in S} M_i, M_S \setminus I_C, K_{S \setminus I_C} \right) \]

(17)

\[= 0, \quad \text{(18)} \]

where the equality (17) follows from (16) and the equality (18) follows from

\[H \left( \sum_{i \in I_C} M_i \big| \sum_{i \in S} M_i, M_S \setminus I_C \right) = 0. \]

We thus have proved the equality (14).

Next, we combine (7), (13) and (14) to obtain

\[\ell \cdot \log q = H \left( \sum_{i \in I_C} M_i \right) = H \left( \sum_{i \in I_C} M_i \big| Y_W, M_S \setminus I_C, K_{S \setminus I_C} \right) \]

\[= H \left( \sum_{i \in I_C} M_i \big| Y_W, M_S \setminus I_C, K_{S \setminus I_C} \right) \]

\[= H \left( \sum_{i \in I_C} M_i \big| Y_C \setminus W, Y_W, M_S \setminus I_C, K_{S \setminus I_C} \right) \]

\[\leq H \left( Y_C \setminus W \big| Y_W, M_S \setminus I_C, K_{S \setminus I_C} \right) \]

\[\leq H \left( Y_C \setminus W \leq |C \setminus W| \cdot \log q \right) \]

\[= n \cdot \left( |C| - |W| \right) \cdot \log q, \quad \text{(19)} \]

i.e.,

\[\frac{\ell}{n} \leq |C| - |W|. \]

Since the inequality (19) holds for all the wiretap sets \(W \subseteq C\) that satisfies \(0 \leq |W| \leq r\) and \(D_W \subseteq I_C\), we obtain that

\[\frac{\ell}{n} \leq \min_{W \subseteq C} \left( |C| - |W| \right). \]

By considering all the cut sets \(C \in A(N)\), we further obtain

\[\frac{\ell}{n} \leq \min_{(W,C) \in W_I \times A(N)} \left( |C| - |W| \right). \]

Finally, since the above upper bound is valid for all the admissible secure network codes for the model \((N, f, r)\), we have proved (6), and hence the theorem is proved.

Remark 1: The upper bound on \(\tilde{C}(N, f, r)\) in Theorem 1 is always nonnegative because for any \((W, C)\) satisfying the condition in the minimum, we have \(W \subseteq C\), and so \(|C| \geq |W|\).

Corollary 2: Let \(C_{\min} = \min_{1 \leq i \leq s} \text{mincut}(\sigma_{i, \rho})\). The upper bound on the secure computing capacity \(\tilde{C}(N, f, r)\) obtained in Theorem 1 is bounded by

\[C_{\min} - r \leq \min_{(W,C) \in W_I \times A(N)} \left( |C| - |W| \right) \leq C_{\min}. \quad \text{(20)} \]

Proof: We first prove the lower bound in (20). We consider

\[\min_{(W,C) \in W_I \times A(N)} \left( |C| - |W| \right) \]

\[= \min_{W \subseteq C \cap D_W \subseteq I_C} \left( |C| - |W| \right) \]

\[= \min_{W \subseteq C \cap D_W \subseteq I_C} \max_{W \subseteq C \cap D_W \subseteq I_C} \left( |C| - |W| \right) \]

\[\geq \min_{C \in A(N)} |C| - r \]

\[= C_{\min} - r, \quad \text{(21)} \]

where the inequality (21) follows from \(|W| \leq r, \forall W \in W_I\); and the equality (22) follows from the observation that

\[C_{\min} = \min_{C \in A(N)} \left\{ |C| : C \in A(N) \right\}, \quad \text{(22)} \]

which is explained as follows. First, we have

\[C_{\min} \geq \min_{C \in A(N)} \left\{ |C| : C \in A(N) \right\}. \quad \text{(23)} \]

because any minimum cut separating \(\rho\) from a source node \(\sigma_i\) is a cut set in \(A(N)\). On the other hand, consider a cut set \(C^* \in \arg \min_{C} \left\{ |C| : C \in A(N) \right\}\).

Clearly, \(C^*\) is a cut separating \(\rho\) from a source node \(\sigma_i\) in \(I_{C^*}\). This implies

\[|C^*| \geq \text{mincut}(\sigma_i, \rho) \geq C_{\min}. \]

Together with (24), we have proved (23).

Next, we prove the upper bound in (20). Rewrite the upper bound (6) in Theorem 1 as

\[\min_{(W,C) \in W_I \times A(N)} \left( |C| - |W| \right) \]

\[= \min_{W \subseteq C \cap D_W \subseteq I_C} \left( |C| - |W| \right). \quad \text{(25)} \]

Consider the empty set \(\emptyset\), a particular wiretap set in \(W_I\). Clearly, each cut set \(C \in A(N)\) satisfies \(\emptyset \subseteq C\) and \(\emptyset \subseteq D_W \subseteq I_C\). By (25), we immediately obtain that

\[\min_{W \subseteq C \cap D_W \subseteq I_C} \left( |C| - |W| \right) \]

\[\leq \min_{C \in A(N)} \left( |C| - |\emptyset| \right) = C_{\min}. \]

Combining (21) and (25), the corollary is thus proved.

Next, we give two corollaries, which are two cases such that the secure computing capacities are exactly characterized by the upper bound in Theorem 1.

Corollary 3: Consider the model \((N, f, r)\), where the target function \(f\) is the algebraic sum over a finite field \(\mathbb{F}_q\). If the security level \(r\) satisfies

\[r \geq \tilde{C}_{\min} \triangleq \min_{C \in A(N)} \left\{ |C| : C \in A(N) \right\}, \quad \text{(26)} \]

then

\[\tilde{C}(N, f, r) = 0. \quad \text{(27)} \]

Proof: Let \(W\) be an edge subset in the set

\[\arg \min_{C} \left\{ |C| : C \in A(N) \right\}. \quad \text{(27)} \]

Since \(I_W \neq \emptyset\), we readily see that \(W \in A(N)\). On the other hand, by (26) we have \(|W| \leq r\) and so \(W\) is a wiretap set in \(W_I\). Together with \(D_W = I_W\) from (27), \((W, W)\) is a pair.
in $\mathcal{W}_r \times \Lambda(\mathcal{N})$ satisfying $W \subseteq W$ and $D_W \subseteq I_W$. Then, the upper bound (6) obtained in Theorem 1 is 0, which, together with $\tilde{C}(N, f, r) \geq 0$, implies $\tilde{C}(N, f, r) = 0$.

A straightforward consequence of Corollary 3 is that $\tilde{C}(N, f, r) = 0$ if $r \geq \min_{1 \leq i \leq m} |\text{Out}(\sigma_i)|$, because $\text{Out}(\sigma_i)$ satisfies $D_{\text{Out}(\sigma_i)} = I_{\text{Out}(\sigma_i)} \neq \emptyset$ for all $i$. Next, we consider the case of security level $r = 0$. By Theorem 1, we immediately obtain that

$$\tilde{C}(N, f, 0) \leq \min_{W \subseteq C \cap D_W \subseteq I_C} |C| = C_{\min}.$$  \hfill (28)

In fact, $C_{\min}$ is the computing capacity for the algebraic sum over an arbitrary network without security consideration (cf. [13], [14]). Thus, the equation (28) shows that the upper bound (6) thus obtained in Theorem 1 degenerates to the computing capacity, denoted by $C(N, f)$, for the algebraic sum $f$ over a network $N$, which is stated in the following corollary.

**Corollary 4:** When the security level $r = 0$,

$$\tilde{C}(N, f, 0) = C(N, f) = C_{\min}.$$  

However, we claim that the security level $r = 0$ is not necessary for our upper bound to be equal to $C_{\min}$. In other words, the secure computing capacity may still achieve $C_{\min}$ even though the security level $r$ is strictly larger than 0. We illustrate this point by Example 1 below, which reveals the surprising fact that for some models of secure network function computation, there is no penalty on the secure computing capacity compared with the computing capacity without security consideration. An interesting open question here is that why and under what conditions the secure computing capacity achieves $C_{\min}$ with a security level strictly larger than 0.

**Example 1:** We consider securely computing the algebraic sum $f(m_1, m_2) = m_1 + m_2$ on $\mathbb{F}_2$ over the network $N_1$ depicted in Fig. 1 with the security level $r = 1$. For this secure model $(N_1, f, 1)$, by Corollary 2 we have

$$\tilde{C}(N_1, f, 1) \leq C_{\min} = 1.$$  

On the other hand, we present an admissible $(1, 1)$ secure network code in Fig. 1, of which the secure computing rate achieves the upper bound $C_{\min} = 1$. Here, $k_i \in \mathcal{K}_i = \mathbb{F}_2$ is an arbitrary output of the random key $K_i$ for the source node $\sigma_i$, $i = 1, 2$.

**B. An Efficient Approach for Computing the Upper Bound**

In this subsection, we will provide an efficient approach for computing the upper bound (6) in Theorem 1. We consider a wiretap set $W \in \mathcal{W}_r$ and define

$$\Omega(W) = \min_{W \subseteq C \cap D_W \subseteq I_C} (|C| - |W|) = \min_{W \subseteq C \cap D_W \subseteq I_C} |C \setminus W|. \hfill (29)$$

Immediately, the upper bound in Theorem 1 (cf. (6)) can be rewritten as

$$\tilde{C}(N, f, r) \leq \min_{W \in \mathcal{W}_r} \Omega(W).$$

We first extend the definition of a cut separating a subset of nodes $V$ from another subset of nodes $U$ in the graph $G$ to the definition of a cut separating a subset of edges $W$ from a subset of nodes $U$ as follows. We subdivide each edge $e \in W$ by creating a node $v_e$, and splitting $e$ into two edges $e^1$ and $e^2$ such that $\text{tail}(e^1) = \text{tail}(e)$, $\text{head}(e^2) = \text{head}(e)$, and $\text{head}(e^1) = \text{tail}(e^2) = v_e$. Let $W_V = \{v_e : e \in W\}$. Then a cut separating the edge subset $W$ from the set of nodes $U$ is defined by a cut separating $W_V$ from $U$, where, whenever $e^1$ or $e^2$ appears in the cut, replace it by $e$. By definition, we note that $W$ itself is a cut separating $W$ from $U$. Similarly, the minimum cut capacity separating $V_W$ from $U$ is defined as the minimum cut capacity separating $W$ from $U$. Also, a cut separating $W$ from $U$ achieving this minimum cut capacity is called a minimum cut separating $W$ from $U$. Furthermore, we say that a minimum cut separating $W$ from $U$ is primary if it separates from $U$ all the minimum cuts that separate $W$ from $U$. In particular, for a node $v \in V \setminus U$, a primary minimum cut separating $\text{In}(v)$ from $U$ will simply be referred to as a primary minimum cut separating $v$ from $U$. Similarly, we say that an edge subset $W \subseteq E$ is primary if $W$ is the primary minimum cut separating $W$ itself from $D_W$ in the graph $G$. The concept of primary minimum cut was introduced by Guand and Yeung [4], where its existence and uniqueness were proved. We now use a graph in [25] (Fig. 2) as an example to illustrate the above graph-theoretic concepts.

**Example 2:** We consider a node subset $U = \{u_1, u_2\}$ and an edge subset $W = \{e_7, e_8\}$ in the graph $G$ depicted in Fig. 2. For edge $e_7$, we first create a node $v_{e_7}$ and split $e_7$ into two edges $e_7^1$ and $e_7^2$ with $\text{tail}(e_7^1) = \text{tail}(e_7)$, $\text{head}(e_7^2) = \text{head}(e_7)$, and $\text{head}(e_7^1) = \text{tail}(e_7^2) = v_{e_7}$. The same subdivision operation is applied to edge $e_8$ as depicted in Fig. 3. Let $W_V = \{v_{e_7}, v_{e_8}\}$. Now, in order to find a cut separating $W$ from $U$, it is equivalent to finding a cut separating $W_V$ from $U$. By definition, we see that the edge subset $\{e_3, e_4\}$ is a cut separating $W$ from $U$, because no path exists from $u_1$ or $u_2$ to $v_{e_7}$ or $v_{e_8}$ upon removing $\{e_3, e_4\}$. 

![Fig. 1. An admissible (1,1) secure network code for the secure model (N_1, f, 1).](image-url)
Also, the edge subset \( \{ e_1 \} \) is a cut separating \( V_W \) from \( U \). By definition, \( e_1 \) appears in the cut \( \{ e_1 \} \) and \( e_7 \in W \), and thus \( e_1 \) is replaced by \( e_7 \) and so \( \{ e_7 \} \) is a cut separating \( W' \) from \( U \).

We further see that \( \{ e_7 \} \) is a minimum cut separating \( W \) from \( U \) that achieves the minimum cut capacity separating \( W \) from \( U \) whose value is 1.

Furthermore, we see that \( \{ e_5 \} \) is also a minimum cut separating \( W \) from \( U \), and \( \{ e_5 \} \) and \( \{ e_7 \} \) are all the minimum cuts separating \( W \) from \( U \). We further observe that \( \{ e_5 \} \) separating from \( U \) all the minimum cuts separating \( W \) from \( U \), i.e., two singleton subsets of edges \( \{ e_5 \} \) and \( \{ e_7 \} \). Therefore, by definition, \( \{ e_5 \} \) is the primary minimum cut separating \( W \) from \( U \).

Next, we prove several lemmas that are instrumental in establishing Theorem 9.

**Lemma 5:** Let \( W \) be an edge subset and \( W' \) be a minimum cut separating \( W \) from \( D_W \). Then,

\[
D_W = D_W'.
\]

**Proof:** First, since \( W' \) is a cut separating \( W \) from \( D_W \), namely that any path from a source node in \( D_W \) to an edge in \( W \) must pass through an edge in \( W' \), we obtain that \( D_W \subseteq D_W' \). It thus suffices to prove that \( D_W \subseteq D_W \). This can be proved by contradiction as follows. Assume that there exists a source node \( e \in D_W \) but \( e \notin D_W' \). Let \( e \) be an edge in \( W' \) such that \( e \rightarrow e \). Then, \( e \rightarrow d, \forall d \in W \), because otherwise there exists an edge \( d \in W \) such that \( e \rightarrow d \), which, together with \( e \rightarrow e \), implies \( e \rightarrow d \), i.e., \( e \in D_W \). This is a contradiction to the assumption that \( e \notin D_W \).

Now, we remove the edge \( e \) from \( W' \) to obtain a new edge subset \( W' \setminus \{ e \} \). By \( e \rightarrow d \) for all \( d \in W \) as discussed above, we see that the edge subset \( W' \setminus \{ e \} \) also separates \( W \) from \( D_W \). This contradicts the assumption that \( W' \) is a minimum cut separating \( W \) from \( D_W \). The lemma is proved.

In the rest of the paper, for an edge subset \( W \subseteq \mathcal{E} \), we let \( W \) be the primary minimum cut separating \( W \) from \( D_W \). Then, \( W \) is primary if and only if \( \Omega(W) \).

**Lemma 6:** Consider a wiretap set \( W \subseteq \mathcal{E} \). Let \( W' \) be a minimum cut separating \( W \) from \( D_W \). Then,

\[
\Omega(W') \leq \Omega(W) \leq \Omega(W').
\]

**Proof:** Let \( W \in W' \) and let \( C_W \in \Lambda(N) \) be an arbitrary but fixed subset of edges in

\[
\arg \min_{C \in \Lambda(N)} \{ |C| : W \subseteq C \text{ and } D_W \subseteq I_C \}.
\]

By (29), we readily see that

\[
O(W) = |C_W \setminus W|.
\]

Let \( W' \) be a minimum cut separating \( W \) from \( D_W \), and let

\[
C' = W' \cup (C_W \setminus W).
\]

We claim that \( C' \) is a cut separating the sink node \( \rho \) from \( D_W \). To see this, assume the contrary that there exists a path \( \rho \) from a source node \( e \in D_W \) to \( \rho \) that does not pass through any edge in \( C' \), or equivalently, the path \( \rho \) does not pass through any edge in \( W' \) or \( C_W \setminus W \). Note that \( C_W \) is a cut separating \( \rho \) from \( I_C \), and by (30) we have \( D_W \subseteq I_C \). This immediately implies that \( C_W \) is also a cut separating \( \rho \) from \( D_W \). So, the path \( \rho \) must pass through an edge in \( C_W \), or equivalently, in \( W' \) or \( C_W \setminus W \). Together with \( C_W \subseteq C' \) and the assumption that the path \( \rho \) does not pass through any edge in \( C' \), the path \( \rho \) must pass through an edge in \( W' \). Further, we see that the path \( \rho \) does not pass through any edge in \( W' \) either, because \( \rho \) does not pass through any edge in \( C' \) either. This contradicts the fact that \( W' \) is a cut separating \( W \) from \( D_W \). We thus have proved that \( C' \) is a cut separating \( \rho \) from \( D_W \). This immediately implies \( D_W \subseteq I_C \). Together with \( D_W = D_W \) by Lemma 5, we see that \( C' \) is a cut set in \( \Lambda(N) \) such that \( W' \subseteq C' \) and \( D_W \subseteq I_C \). Consequently, we obtain that

\[
\Omega(W') \leq |C' \setminus W'| = |C' - |W'|
\]

\[
= |W' \cup (C_W \setminus W)| - |W'|
\]

\[
\leq |W'| + |C_W \setminus W| - |W'| = |C_W \setminus W|
\]

\[
= \Omega(W).
\]

Furthermore, by definition, the primary minimum cut \( \hat{W} \) separating \( W \) from \( D_W \) is the common minimum cut from \( D_W \) separating all the minimum cuts that separate \( W \) from \( D_W \). Thus, in the above discussion, by replacing \( W \)
and \( W' \) by \( W \) and \( \widehat{W} \), respectively, we obtain that \( \Omega(\widehat{W}) \leq \Omega(W') \). The lemma is thus proved.

For a set of edges \( W \subseteq \mathcal{E} \), let \( \mathcal{G}_W \) be the residual graph upon deleting the edges in \( W \) from the graph \( \mathcal{G} \).

**Lemma 7:** For a wiretap set \( W \in \mathcal{W}_r \), let \( C^*_W \) be the primary minimum cut separating \( \rho \) from \( D_W \) in the residual graph \( \mathcal{G}_W \). Then

\[
|C^*_W| = \Omega(|W|).
\]

**Proof:** Fix a wiretap set \( W \in \mathcal{W}_r \), and consider an arbitrary cut set \( C \in \Lambda(\mathcal{N}) \) satisfying \( W \subseteq C \) and \( D_W \subseteq I_C \). We readily see that \( C \) is a cut in the graph \( \mathcal{G} \) separating \( \rho \) from \( D_W \). Next, we claim that \( C \setminus W \) is a cut separating \( \rho \) from \( D_W \) in the residual graph \( \mathcal{G}_W \). To see this, assume the contrary that in \( \mathcal{G}_W \) there exists a path \( P \) from a source node \( \sigma \in D_W \) to \( \rho \) that does not pass through any edge in \( C \setminus W \). Then, \( P \) is a path in \( \mathcal{G} \) from \( \sigma \) to \( \rho \) that does not pass through any edge in \( C \setminus W \) or \( W \), i.e., in \( \mathcal{G} \). This contradicts the fact mentioned above that \( C \) is a cut in \( \mathcal{G} \) separating \( \rho \) from \( D_W \). Let \( C^*_W \) be the primary minimum cut separating \( \rho \) from \( D_W \) in \( \mathcal{G}_W \). We immediately have

\[
|C^*_W| \leq |C \setminus W| = |C| - |W|.
\]

Considering all cut sets \( C \in \Lambda(\mathcal{N}) \) satisfying \( W \subseteq C \) and \( D_W \subseteq I_C \), by (31), we have

\[
|C^*_W| \leq \min_{W \subseteq C \text{ and } D_W \subseteq I_C} (|C| - |W|).
\]

On the other hand, for a cut \( C_W \) in \( \mathcal{G}_W \) that separates \( \rho \) from \( D_W \), we see that \( C \triangleq C_W \cup W \) is a cut in \( \mathcal{G} \) that separates \( \rho \) from \( D_W \), because otherwise there exists a path in \( \mathcal{G} \) from a source node \( \sigma \in D_W \) to \( \rho \) that does not pass through any edge in \( C_W \) or \( W \), or equivalently, there exists a path in \( \mathcal{G}_W \) from \( \sigma \) to \( \rho \), which contradicts the fact that \( C_W \) is a cut separating \( \rho \) from \( D_W \) in \( \mathcal{G}_W \). In particular, we consider the primary minimum cut \( C^*_W \) separating \( \rho \) from \( D_W \) in \( \mathcal{G}_W \), and let \( C^* \triangleq C^*_W \cup W \). By the above discussion, \( C^* \) is a cut in \( \Lambda(\mathcal{N}) \) satisfying \( W \subseteq C^* \) and \( D_W \subseteq I_{C^*} \). Thus, we have

\[
|C^*_W| = |C^*| - |W| \geq \min_{W \subseteq C \text{ and } D_W \subseteq I_C} (|C| - |W|),
\]

where the first equality in (33) follows from \( C^*_W \cap W = \emptyset \). Then the lemma is proved by combining (32) and (33).

An important consequence of Lemmas 5 and 6 is given in the following lemma. This, together with Lemma 7, provides an equivalent expression of the upper bound in Theorem 1 which can be used to compute the obtained upper bound efficiently.

Recall that an edge subset \( W \subseteq \mathcal{E} \) is primary if \( W \) is the primary minimum cut separating \( W \) itself from \( D_W \) in the graph \( \mathcal{G} \), or equivalently, \( W = \widehat{W} \).

**Lemma 8:** Let \( \mathcal{W}_r' \subseteq \mathcal{W}_r \) be the collection of all the primary wiretap sets in \( \mathcal{W}_r \), i.e.,

\[
\mathcal{W}_r' = \{ W \in \mathcal{W}_r : W = \widehat{W} \}.
\]

Then,

\[
\min_{W \in \mathcal{W}_r'} \Omega(W) = \min_{W \in \mathcal{W}_r} \Omega(W).
\]

**Proof:** By Lemma 6, we immediately obtain

\[
\min_{W \in \mathcal{W}_r} \Omega(W) \geq \min_{W \in \mathcal{W}_r} \Omega(\widehat{W}).
\]

For each \( W \in \mathcal{W}_r \), since \( |\widehat{W}| \leq |W| \), we have \( \widehat{W} \in \mathcal{W}_r \), and so \( \Omega(\widehat{W}) \geq \min_{W \in \mathcal{W}_r} \Omega(W) \). Thus, \( \min_{W \in \mathcal{W}_r} \Omega(W) \geq \min_{W \in \mathcal{W}_r} \Omega(\widehat{W}) \). Together with (35), we obtain

\[
\min_{W \in \mathcal{W}_r} \Omega(W) = \min_{W \in \mathcal{W}_r} \Omega(\widehat{W}).
\]

To prove (34), it suffices to show \( \min_{W \in \mathcal{W}_r} \Omega(\widehat{W}) = \min_{W \in \mathcal{W}_r} \Omega(W) \), which follows from the claim that for any wiretap set \( W' \in \mathcal{W}_r \), if there exists a \( W \in \mathcal{W}_r \) such that \( W' = \widehat{W} \), then \( W' = \widehat{W} \). We now justify this claim.

By Lemma 5, we have \( \mathcal{W}_r' = \mathcal{W}_r \). Then, \( W' \) is the primary minimum cut separating \( \rho \) from \( D_{W'} \). We now show that \( W' \) is also the primary minimum cut separating itself from \( D_{W'} \), i.e., \( W' = \widehat{W'} \). If the contrary is true, we have \( W' \neq \widehat{W} \).

Then \( \widehat{W} \) is also a minimum cut separating \( \rho \) from \( D_{W'} \), which is linear in \( \mathcal{W}_r \). This is formally stated in the following theorem.

**Theorem 9:** Consider the model of secure network function computation \( (\mathcal{N}, f, \rho) \), where the target function \( f \) is the algebraic sum over a finite field \( \mathbb{F}_q \). The upper bound on the secure computing capacity \( C(\mathcal{N}, f, \rho) \) in Theorem 1 can be rewritten as

\[
C(\mathcal{N}, f, \rho) \leq \min_{W \in \mathcal{W}_r} |C^*_W|,
\]

where \( C^*_W \) is the primary minimum cut separating \( \rho \) from \( D_W \) in the residual graph \( \mathcal{G}_W \).

Theorem 9 induces an efficient approach for computing the upper bound in Theorem 1. To be specific, in order to compute the upper bound in Theorem 1, it suffices to find the primary minimum cut separating \( \rho \) from \( D_W \) in the residual graph \( \mathcal{G}_W \) for each primary wiretap set \( W \in \mathcal{W}_r \). Guang and Yeung [4] developed an efficient algorithm for finding the primary minimum cut separating a node from another one in a directed acyclic graph, and also proved that the computational complexity of this algorithm is linear in the number of the edges in the graph. This algorithm can be straightforwardly extended to a linear-time algorithm for finding the primary minimum cut separating a node from a subset of nodes. So, for each primary minimum cut \( W \) in \( \mathcal{W}_r \), we can find the primary minimum cut \( C^*_W \) separating \( \rho \) from \( D_W \) in the residual graph \( \mathcal{G}_W \) in \( \mathcal{O}(|\mathcal{E}|) \) time. Furthermore, by using the approach in [25, Algorithm 1], we can determine \( \mathcal{W}_r \), or equivalently, find all the primary wiretap sets in \( \mathcal{W}_r \) in \( \mathcal{O}(|\mathcal{W}_r'| \cdot |\mathcal{E}|) \) time (cf. [25, Algorithm 1]). Combining these discussions, we can compute the upper bound in \( \mathcal{O}(|\mathcal{W}_r'| \cdot |\mathcal{E}|) \) time, which is linear in \( |\mathcal{E}| \).
IV. LINEAR (FUNCTION-COMPUTING) SECURE NETWORK CODING

In this section, we will discuss linear secure network coding for computing the algebraic sum. We will present a construction of admissible linear secure network codes for the secure model \((N, f, r)\), which immediately leads to a lower bound on the secure computing capacity \(\overline{C}(N, f, r)\), where \(f\) is defined in (5).

A. Linear Secure Network Coding for Computing the Algebraic Sum

We will define linear (function-computing) secure network codes for the secure model \((N, f, r)\) in this subsection. Briefly speaking, a secure network code is said to be linear if its local encoding function for each edge is linear. Let \(\ell\) and \(n\) be two positive integers. We now define an \((\ell, n)\) linear secure network code over the finite field \(\mathbb{F}_q\) for the secure model \((N, f, r)\). Each source node \(\sigma_i\) sequentially generates \(\ell\) i.i.d. random variables \(M_{i,1}, M_{i,2}, \ldots, M_{i,\ell}\) with generic random variable \(M_i\). We let \(M_i = (M_{i,1}, M_{i,2}, \ldots, M_{i,\ell})\) and \(M_S = (M_1, M_2, \ldots, M_s)\). Furthermore, we continue to use \(K_i\) to denote the random key available at the source node \(\sigma_i\), which is now assumed to be distributed uniformly on the vector space \(\mathbb{F}_q^n\), where \(r_i\) is a nonnegative integer. Let \(K_S = (K_1, K_2, \ldots, K_s)\).

Now, for \(i = 1, 2, \ldots, s\), we let \(m_i \in \mathbb{F}_q^\ell\) and \(k_i \in \mathbb{F}_q^n\) be arbitrary outputs of the source message \(M_i\) and the key \(K_i\), respectively. Accordingly, let \(m_S = (m_1, m_2, \ldots, m_s)\) and \(k_S = (k_1, k_2, \ldots, k_s)\), i.e., \(m_S = (m_1, m_2, \ldots, m_s)\) and \(k_S = (k_1, k_2, \ldots, k_s)\). Now, let \(x_i = (m_i, k_i)\) for \(i = 1, 2, \ldots, s\) and let \(x_S = (x_1, x_2, \ldots, x_s)\).

An \((\ell, n)\) linear secure network code \(\tilde{C} = \{\tilde{\sigma}_e : e \in E; \tilde{\phi}\}\) consists of a linear local encoding function \(\tilde{\theta}_e\) for each edge \(e\) and a decoding function \(\tilde{\phi}\) at the sink node \(\rho\) as follows:

- For each local linear encoding function \(\tilde{\theta}_e\),
  \[
  \tilde{\theta}_e : \begin{cases}
  \mathbb{F}_q^{\ell} \times \mathbb{F}_q^n \to \mathbb{F}_q^n, & \text{if tail}(e) = \sigma_i \text{ for some } i, \\
  \prod_{d \in \text{In}(\text{tail}(e))} \mathbb{F}_q \to \mathbb{F}_q^n, & \text{otherwise};
  \end{cases}
  \tag{36}
  \]
  or more precisely,
  \[
  \tilde{\theta}_e(x_i) = x_i \cdot A_{i,e} = (m_i, k_i) \cdot A_{i,e}, \quad \text{if tail}(e) = \sigma_i \text{ for some } i,
  \]
  \[
  \tilde{\theta}_e(y_d : d \in \text{In}(v)) = \sum_{d \in \text{In}(v)} y_d \cdot A_{d,e}, \quad \text{otherwise};
  \tag{37}
  \]

where \(v \triangleq \text{tail}(e)\) is a node in \(V \setminus \{S \cup \rho\}\). \(A_{i,e}\) is an \(\mathbb{F}_q\)-valued matrix of size \((\ell + r_i) \times n\), \(y_d \in \mathbb{F}_q^n\) is a row vector standing for the message transmitted on an edge \(d\), and \(A_{d,e}\) is an \(\mathbb{F}_q\)-valued matrix of size \(n \times n\) called the local encoding matrix of the adjacent edge pair \((d, e)\) for the linear secure network code \(\tilde{C}\).

- The decoding function \(\tilde{\phi}\) at the sink node \(\rho\) is a mapping from \(\prod_{d \in \text{In}(\rho)} \mathbb{F}_q^n\) to \(\mathbb{F}_q^n\), which is used to compute at \(\rho\) the algebraic sum \(f\) with zero error.

With the linear encoding mechanism as described in (37), we readily see that \(y_e\) for each \(e \in E\) is a linear function of \(m_S\) and \(k_S\), i.e., each global encoding function \(\tilde{g}_e\), induced by the linear local encoding functions \(\tilde{\theta}_e\), is linear. Then, for each \(e \in E\), there exists an \(\mathbb{F}_q\)-valued matrix \(g_e\) of size \((\ell + r_i + 1) \times n\) such that

\[
\tilde{g}_e(x_S) = x_S \cdot g_e = y_e.
\]

In the rest of the paper we use \((m_S, k_S)\) to represent \((m_1, k_1) (m_2, k_2) \cdots (m_s, k_s)\), i.e., \((m_S, k_S) = x_S\), for the convenience of discussion. This abuse of notation should cause no ambiguity and would greatly simplify the notation. Note that the \((\ell, n)\) linear secure network code as defined is sometimes referred to in the literature (e.g., [26], [27], [28], [29]) as a “vector-linear” code when \(n > 1\).

We end this subsection with two remarks on the definition of linear secure network codes.

Remark 2: In the definition of linear secure network codes, the decoding function \(\tilde{\phi}\) is not necessarily linear. Nevertheless, it can be proved that for any admissible linear secure network code \(\tilde{C} = \{\tilde{\theta}_e : e \in E; \tilde{\phi}\}\) for computing the algebraic sum, there always exists a linear decoding function \(\tilde{\phi}'\) such that the code \(\{\tilde{\theta}_e : e \in E; \tilde{\phi}'\}\) is still admissible.

Remark 3: If \(r_i = 0\) in (36) for all \(i = 1, 2, \ldots, s\), then the definition of linear secure network codes for the secure model \((N, f, r)\) degenerates to the definition of linear network codes for the model \((N, f)\).

We say a linear network code for \((N, f)\) is admissible if the target function \(f\) can be computed at the sink node with zero error, i.e., only the computability condition is satisfied.

B. Lower Bound on the Secure Computing Capacity

We consider the secure model \((N, f, r)\) with security level \(0 \leq r \leq C_{\min}\), where we recall that

\[
C_{\min} \triangleq \min_{1 \leq i \leq s} \text{mincut}(\sigma_i, \rho).
\]

For any nonnegative integer \(R\) with \(r \leq R \leq C_{\min}\), we can construct an admissible \((R - r, 1)\) linear secure network code for the model \((N, f, r)\). This code construction immediately provides a lower bound on the secure computing capacity \(\overline{C}(N, f, r)\), which is stated in the following theorem. The proof of the theorem, including the code construction and the verification of the computability and security conditions, is deferred to Sections IV-C and IV-D.

Theorem 10: Consider the model of secure network function computation \((N, f, r)\), where the target function \(f\) is the algebraic sum over a finite field \(\mathbb{F}_q\) and the security level \(r\) satisfies \(0 \leq r \leq C_{\min}\). Then

\[
\overline{C}(N, f, r) \geq C_{\min} - r.
\]

We recall from (23) and (26) that

\[
C_{\min} = \min \{|C| : C \in \Lambda(N)\},
\]

and

\[
\overline{C}_{\min} \triangleq \min \{|C| : C \in \Lambda(N) \text{ and } D_C = I_C\}. \tag{38}
\]
Clearly, we have \( C_{\min} \leq \overline{C}_{\min} \). By applying the lower bound in Theorem 10, the secure computing capacity \( \hat{C}(N, f, r) \) can be fully characterized when \( C_{\min} = \overline{C}_{\min} \), i.e., for some source node \( \sigma_i \in S \), there exists a minimum cut \( C \) separating \( \rho \) from \( \sigma_i \) such that \( |C| = C_{\min} \) and \( D_C = I_C \). This is given in the next corollary.

**Corollary 11:** When \( C_{\min} = \overline{C}_{\min} \),
\[
\hat{C}(N, f, r) = \begin{cases} 
C_{\min} - r, & \text{if } r < C_{\min}, \\
0, & \text{otherwise}.
\end{cases}
\]

**Proof:** First, by Corollary 3 we have
\[
\hat{C}(N, f, r) = 0, \quad \forall \ r \geq C_{\min}.
\]

When \( C_{\min} = \overline{C}_{\min} \), it immediately follows from (39) that \( \hat{C}(N, f, r) = 0 \) for \( r \geq C_{\min} \).

Next, we consider the case of \( r < C_{\min} \). As \( C_{\min} = \overline{C}_{\min} \), in light of (38), there exists a cut set \( C \in \Lambda(N) \) such that \( D_C = I_C \) and \( |C| = C_{\min} \). Now, we let \( \hat{W} \) be an edge subset of \( C \) with \( |\hat{W}| = r \). Immediately, we see that \( \hat{W} \subseteq W_r \) and \( D_{\hat{W}} \subseteq D_C = I_C \). By the upper bound (6) in Theorem 1, we have
\[
\hat{C}(N, f, r) \leq \min_{\substack{\{W_C \subseteq W \mid |W_C| = |\hat{W}| \}}}( |C| - |W| ) \\
\leq |C| - |\hat{W}| = C_{\min} - r.
\]

Together with the lower bound in Theorem 10, we have proved that \( \hat{C}(N, f, r) = C_{\min} - r \) for \( r < C_{\min} \).

The corollary is thus proved.

When \( C_{\min} < \overline{C}_{\min} \), by Corollary 3, \( \hat{C}(N, f, r) = 0 \) if \( r \geq C_{\min} \). When \( C_{\min} < \overline{C}_{\min} \) and the security level \( r \) satisfies \( r \leq C_{\min} \), we give in the next corollary a sufficient condition for the network topology in Theorem 10. However, for \( C_{\min} < r < \overline{C}_{\min} \), a nontrivial lower bound\(^3\) on \( \hat{C}(N, f, r) \) or a code construction is yet to be obtained.

**Corollary 12:** When \( C_{\min} < \overline{C}_{\min} \),
\[
\hat{C}(N, f, r) = C_{\min} - r, \quad \forall \ r \leq C_{\min},
\]

provided that for some source node \( \sigma_i \in S \) with \( \min cut(\sigma_i, \rho) = C_{\min} \), there exists a minimum cut \( C \) separating \( \rho \) from \( \sigma_i \) such that there are \( r \) edges in \( C \) satisfying that for each such edge \( e \),
\[
\sigma \rightarrow e, \quad \forall \ \sigma \in S \setminus I_C. \quad \text{(4)}
\]

**Proof:** Let \( \sigma_i \) be a source node in \( S \) such that
1) \( \min cut(\sigma_i, \rho) = C_{\min} \), and
2) there exists a minimum cut \( C \) separating \( \rho \) from \( \sigma_i \) such that \( C \) edges in \( C \) satisfying that for each edge \( e \) of the \( r \) edges, \( \sigma \rightarrow e, \ \forall \ \sigma \in S \setminus I_C. \)

Let \( W \) be any subset of \( r \) edges in \( C \) satisfying the above condition 2). Then, we have
\[
W \subseteq C \quad \text{and} \quad D_W \subseteq I_C.
\]

By \( W \in W_r \) (since \( |W| = r \)) and the upper bound in Theorem 1, we thus obtain that
\[
\hat{C}(N, f, r) \leq |C| - |W| = C_{\min} - r.
\]

Together with the lower bound in Theorem 10, we have proved that \( \hat{C}(N, f, r) = C_{\min} - r \).

**C. Code Construction**

Before presenting the code construction for the secure model \( (N, f, r) \) with security level \( 0 \leq r \leq C_{\min} \), we first consider the model \( (N, f) \), i.e., computing the algebraic sum \( f \) over \( N \) without any security constraint. For any nonnegative integer \( R \) with \( r \leq R \leq C_{\min} \), we can construct an admissible \((R, 1)\) linear network code for \((N, f)\) by “reversing” a rate-\( R \) linear network code for the single-source multicast problem on the reversed network \( N^T \triangleq (G^T, S, \rho) \), where the reversed graph \( G^T \) is obtained by \( G \) by reversing the direction of every edge in \( G \) and \( \rho \) and \( S \) are regarded as the single source node and the set of sink nodes, respectively. For the completeness of the paper, we include a discussion on the construction of such an admissible \((R, 1)\) linear network code for \((N, f)\) in Appendix A (also cf. [7], [8]). In the following, we will construct an admissible \((R, 1)\) linear secure network code for the secure model \((N, f, r)\) based on an admissible \((R, 1)\) linear network code for \((N, f)\). We refer the reader to the following example in Section IV-F to illustrate our code construction.

By the acyclicity of the graph \( G \), we can fix a topological order \( \prec \) on the edges in \( E \) that is consistent with the natural partial order of the edges. To facilitate our discussion, we further assume without loss of generality that this order \( \prec \) satisfies the following two conditions: i) the output edges of the source nodes in \( \bigcup_{i=1}^{s} \text{Out}(\sigma_i) \) are prior to the other edges in \( E \setminus \bigcup_{i=1}^{s} \text{Out}(\sigma_i) \), i.e.,
\[
d \prec e, \quad \forall \ d \in \bigcup_{i=1}^{s} \text{Out}(\sigma_i) \quad \text{and} \quad e \in E \setminus \bigcup_{i=1}^{s} \text{Out}(\sigma_i);
\]
and ii) for any two edges \( e_i \in \text{Out}(\sigma_i) \) and \( e_j \in \text{Out}(\sigma_j) \),
\[
e_i \prec e_j \quad \text{if} \quad i < j.
\]

In the rest of the paper, we always use this order to index the coordinates of vectors and the rows/columns of matrices.

Let \( R \) be a positive integer not larger than \( C_{\min} \). Let \( x_i \triangleq (x_{i,1}, x_{i,2}, \ldots, x_{i,R}) \in \mathbb{F}_2^R \) be the vector of \( R \) symbols sequentially generated by the source node \( \sigma_i \) for \( i = 1, 2, \ldots, s \), and let \( x_S = (x_1, x_2, \ldots, x_s) \). Further, let \( C = \{ \theta_e : e \in E; \ \varphi \} \) be an admissible \((R, 1)\) linear network code for computing the algebraic sum \( f \) over the network \( N \), where \( \theta_e \) is the local encoding function of edge \( e \). By the linearity

\(^3\)In Appendix A, we use a global approach to describe the relation between a linear network code and its “reversed” code, which is different from the local approach in [7] and [8]. Nevertheless, the essence of the two approaches are the same.
where \( v = \text{tail}(e), A_{i,e} \) is a column \( R \)-vector in \( \mathbb{F}_q \), is the message transmitted on an edge for the source messages \( x_S \), and \( q_{d,e} \in \mathbb{F}_q \) is the local encoding coefficient of the adjacent edge pair \((d, e)\). We further let \( q_{e}, e \in \mathcal{E} \) be the linear global encoding functions induced by the linear local encoding functions \( \sigma_e, e \in \mathcal{E} \), i.e.,

\[
y_e = q_e(x_S), \quad \forall \ e \in \mathcal{E}.
\]

By the linearity of the global encoding function \( q_e \), there exists an \( \mathbb{F}_q \)-valued column \( \mathbf{R} \)-vector \( \bar{g}_e \) such that

\[
q_e(x_S) = x_S \cdot \bar{g}_e.
\]

We call \( \bar{g}_e \) the global encoding vector for the edge \( e \). We further write

\[
\bar{g}_e = \begin{bmatrix} g^{(\sigma_1)}_e \\ g^{(\sigma_2)}_e \\ \vdots \\ g^{(\sigma_s)}_e \end{bmatrix},
\]

where each \( g^{(\sigma_i)}_e \) is a column \( R \)-vector for \( i = 1, 2, \ldots, s \). Then, the equation (41) can be written as

\[
q_e(x_S) = x_S \cdot \bar{g}_e = \sum_{i=1}^{s} x_i \cdot g^{(\sigma_i)}_e.
\]

Next, we will elaborate on how to calculate the global encoding vectors from the linear local encoding functions. To facilitate our discussion, for each source node \( \sigma_i \), we introduce \( R \) imaginary edges \( \bar{d}_{i,1}^t, \bar{d}_{i,2}, \ldots, \bar{d}_{i,R}^t \) connecting to \( \sigma_i \), and let \( \text{In}(\sigma_i) = \{ \bar{d}_{i,1}^t, \bar{d}_{i,2}, \ldots, \bar{d}_{i,R}^t \} \). We further assume that the \( R \) source messages \( x_{i,j} \), \( 1 \leq j \leq R \) are transmitted to \( \sigma_i \) through these \( R \) imaginary edges. To be specific, without loss of generality, assume that the source message \( x_{i,j} \) is transmitted to \( \sigma_i \) through the \( j \)th imaginary edge \( \bar{d}_{i,j}^t \) for \( j = 1, 2, \ldots, R \), i.e.,

\[
y_{\bar{d}_{i,j}^t} = x_{i,j}, \quad 1 \leq j \leq R,
\]

where \( y_{\bar{d}_{i,j}^t} \) stands for the symbol transmitted on \( \bar{d}_{i,j}^t \). For each source node \( \sigma_i \), we define an \( R \times |\mathcal{E}| \) matrix

\[
A_{\sigma_i} = \begin{bmatrix} A_{\sigma_i,e} : \ e \in \mathcal{E} \end{bmatrix},
\]

where

\[
A_{\sigma_i,e} = \begin{cases} A_{i,e} & \text{if } e \in \text{Out}(\sigma_i) \\ 0 & \text{otherwise} \end{cases}
\]

(cf. (40) for \( A_{i,e} \)). For each intermediate node \( v \in \mathcal{V} \setminus (\{S\} \cup \{\rho\}) \), we define an \( |\text{In}(v)| \times |\text{Out}(v)| \) matrix

\[
A_v = (a_{d,e})_{d \in \text{In}(v), \ e \in \text{Out}(v)},
\]

where \( a_{d,e} \) is the local encoding coefficient of the adjacent edge pair \((d, e)\) (cf. (40)). Further, we extend the local encoding coefficient \( a_{d,e} \) to the case of non-adjacent edge pairs \((d, e)\) by setting \( a_{d,e} = 0 \) if \( \text{head}(d) \neq \text{tail}(e) \). In particular, we have \( a_{\epsilon,e} = 0, \forall \ e \in \mathcal{E} \). Now, we let

\[
A = (a_{d,e})_{d \in \mathcal{E}, e \in \mathcal{E}}
\]

be an \( |\mathcal{E}| \times |\mathcal{E}| \) matrix in which all the entries are well defined. Then \( A \) is upper triangular and all the diagonal elements are equal to zero.

Similar to linear network coding (cf. [30]), we have

\[
\bar{g}_e : \ e \in \mathcal{E} = A_{\sigma_1} \cdot (I - A)^{-1},
\]

where \( I \) is the \( |\mathcal{E}| \times |\mathcal{E}| \) identity matrix. Together with (42), we further have

\[
\bar{g}_e : \ e \in \mathcal{E} = \begin{bmatrix} A_{\sigma_1} : e \in \mathcal{E} \\ A_{\sigma_2} : e \in \mathcal{E} \\ \vdots \\ A_{\sigma_s} : e \in \mathcal{E} \end{bmatrix}. (I - A)^{-1}.
\]

Immediately, for each edge \( e \in \mathcal{E} \), it follows from (46) and (47) that

\[
\bar{g}_{e}^{(\sigma_i)} = A_{\sigma_i} \cdot (I - A)^{-1} \cdot \bar{I}_e, \quad \forall \ i = 1, 2, \ldots, s, \tag{48}
\]

and

\[
\bar{g}_e = \begin{bmatrix} A_{\sigma_1} \\ A_{\sigma_2} \\ \vdots \\ A_{\sigma_s} \end{bmatrix} \cdot (I - A)^{-1} \cdot \bar{I}_e,
\]

where \( \bar{I}_e \) is a column \( |\mathcal{E}| \)-vector whose component indexed by the edge \( e \) is equal to 1 while all other components are equal to 0.

Based on the admissible \((R, 1)\) linear network code \( C = \{ \theta_e : e \in \mathcal{E} ; \phi \} \) for the model \((N, f)\), by using a similar approach as the one in [22], we will construct an admissible \((R - r, 1)\) linear secure network code \( \bar{C} = \{ \theta : e \in \mathcal{E} ; \bar{\phi} \} \) for the secure model \((\bar{N}, \bar{f}, r)\) with any security level \( r \leq R \). First, let \( \bar{b}_1, \bar{b}_2, \ldots, \bar{b}_R \) be \( R \) linearly independent column vectors in \( \mathbb{F}_q^R \) such that the first \( R - r \) vectors \( \bar{b}_1, \bar{b}_2, \ldots, \bar{b}_{R-r} \) satisfy the condition that for all the wiretap sets \( W \in \mathcal{W}_r \) and all \( i = 1, 2, \ldots, s \),

\[
\langle \bar{b}_{j} : 1 \leq j \leq R - r \rangle \cap \langle \bar{g}_{e}^{(\sigma_i)} : e \in W \rangle = \{0\}. \tag{49}
\]

The existence of such \( R \) vectors will be discussed in the following Section IV-E. Then, let

\[
B = \begin{bmatrix} \bar{b}_1 & \bar{b}_2 & \cdots & \bar{b}_R \end{bmatrix},
\]

\( \bar{B}_1 \) denote the vector space spanned by the vectors in a set of vectors \( L \).
an $R \times R$ invertible matrix in $\mathbb{F}_q$, and
\[
\begin{bmatrix}
B & B & \cdots & B
\end{bmatrix}^{\text{diagonal}}
\]
\[
\hat{B} = \begin{bmatrix} B & B & \cdots & B \end{bmatrix}_{\text{diagonal}} = \begin{bmatrix} B & 0 & \cdots & 0 \\
0 & B & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & B \end{bmatrix}_{R \times R}
\]
an $s \times s$ block matrix in which each block in the diagonal is $B$ and all the other blocks are the $R \times R$ zero matrix. Since the matrix $B$ is invertible, $\hat{B}$ is also invertible and
\[
\hat{B}^{-1} = \begin{bmatrix} B^{-1} & B^{-1} & \cdots & B^{-1} \end{bmatrix}^{\text{diagonal}}.
\]

We recall that at the source node $\sigma_i$, the information source $M_i$ is distributed uniformly on the vector space $\mathbb{F}_q$, $i = 1, 2, \ldots, s$. Let $M_i = (M_{i,1}, M_{i,2}, \ldots, M_{i,R-r})$ be the vector of $R-r$ i.i.d. random variables generated by $\sigma_i$ with the generic random variable $M_i$, and $M_S = (M_1, M_2, \ldots, M_s)$. Furthermore, we let $K_i$ be a random variable that is distributed uniformly on $\mathbb{F}_q$. Let the key available at the source node $\sigma_i$ be $K_i = (K_{i,1}, K_{i,2}, \ldots, K_{i,r})$, where $K_{i,1}, K_{i,2}, \ldots, K_{i,r}$ are i.i.d random variables with generic random variable $K_i$. Then, the key $K_i$ can be regarded as a random variable distributed uniformly on the vector space $\mathbb{F}_q^r$. Let $K_S = (K_1, K_2, \ldots, K_s)$. All the keys $K_i$ and the source messages $M_i$, $i = 1, 2, \ldots, s$ are mutually independent. Let $m_i, e_i \in \mathbb{F}_q^R$ and $k_i \in \mathbb{F}_q^R$ be arbitrary outputs of the source message $M_i$ and the key $K_i$, respectively, for $i = 1, 2, \ldots, s$. Accordingly, let $m_S = (m_1, m_2, \ldots, m_s)$ and $k_S = (k_1, k_2, \ldots, k_s)$, and let $x_i = (m_i, k_i) \in \mathbb{F}_q^R$ and $x_S = (x_1, x_2, \ldots, x_s)$. By applying the linear network code $C$ and the matrix $B$, we define the linear local encoding function $\hat{\theta}_e$ for each edge $e \in E$ as follows:

- if $\text{tail}(e) = \sigma_i$ for some $i$, $1 \leq i \leq s$, then
  \[
  \hat{\theta}_e(x_i) = x_i \cdot B^{-1} \cdot A_{i,e} = (m_i, k_i) \cdot B^{-1} \cdot A_{i,e};
  \]
- otherwise,
  \[
  \hat{\theta}_e(y_d, d \in \text{In}(v)) = \hat{\theta}_e(y_d, d \in \text{In}(v)) = \sum_{d \in \text{In}(v)} a_{d,e} \cdot y_d;
  \]
where $v = \text{tail}(e)$, an intermediate node in $V \setminus (S \cup \{\rho\})$, and $y_d \in \mathbb{F}_q$ is the message transmitted on edge $d$ for the source message vector $m_S$ and the key vector $k_S$.

For each $e \in E$, we let $\tilde{h}_e$ be the global encoding vector for the secure network code $\tilde{C}$. Similar to (42), we write
\[
\tilde{h}_e = \begin{bmatrix} \tilde{h}_e^{(\sigma_1)} \\
\tilde{h}_e^{(\sigma_2)} \\
\vdots \\
\tilde{h}_e^{(\sigma_s)} \end{bmatrix}
\]
and by the mechanism of linear network coding (similar to (48)), we have
\[
\tilde{h}_e^{(\sigma_i)} = B^{-1} \cdot A_{i,e} \cdot (I - A)^{-1} \cdot \tilde{y}_e = B^{-1} \cdot \tilde{g}_e^{(\sigma_i)},
\]
\[
\forall i = 1, 2, \ldots, s.
\]

Then, we obtain that
\[
\tilde{h}_e = \begin{bmatrix} \tilde{h}_e^{(\sigma_1)} \\
\tilde{h}_e^{(\sigma_2)} \\
\vdots \\
\tilde{h}_e^{(\sigma_s)} \end{bmatrix} = \begin{bmatrix} B^{-1}A_{\sigma_1} \\
B^{-1}A_{\sigma_2} \\
\vdots \\
B^{-1}A_{\sigma_s} \end{bmatrix} \cdot (I - A)^{-1} \cdot \tilde{y}_e
\]
\[
= \tilde{B}^{-1} \cdot \begin{bmatrix} A_{\sigma_1} \\
A_{\sigma_2} \\
\vdots \\
A_{\sigma_s} \end{bmatrix} \cdot (I - A)^{-1} \cdot \tilde{y}_e = \tilde{B}^{-1} \cdot \tilde{g}_e
\]
\[
(51)
\]

(cf. (50) for $\hat{B}^{-1}$). Thus, we have constructed an $(R-r, 1)$ linear network code $\tilde{C} = \{\theta_e : e \in E; \tilde{C}\}$ for the secure model $(N,f,r)$, where the decoding function $\tilde{C}$ will be clear later. The computability and security conditions (i.e., the admissibility) of the code $\tilde{C}$ will be verified in the next subsection.

D. Verification of the Computability and Security Conditions
Verification of the Computability Condition: At the sink node $\rho$, the messages $y_e, e \in \text{In}(\rho)$ are received. Further, we let
\[
G_\rho = [\tilde{g}_e : e \in \text{In}(\rho)] \quad \text{and} \quad H_\rho = [\tilde{h}_e : e \in \text{In}(\rho)].
\]

Clearly, we have $H_\rho = \tilde{B}^{-1} \cdot G_\rho$ by (51), and then
\[
y_\rho \triangleq (y_e : e \in \text{In}(\rho)) = (x_S \cdot \tilde{h}_e : e \in \text{In}(\rho)) = x_S \cdot H_\rho = x_S \cdot \tilde{B}^{-1} \cdot G_\rho = (x_1 B^{-1} x_2 B^{-1} \cdots x_s B^{-1}) \cdot G_\rho.
\]

Note that in (52), $x_S$ is unknown while $y_\rho$ and $H_\rho$ (or equivalently, $\tilde{B}^{-1} \cdot G_\rho$) are known. We aim to compute $\sum_{i=1}^s m_i$ by “solving” the equation (52).

In light of (52), $y_\rho$ can also be regarded as the received vector at $\rho$ when we consider $x_i B^{-1}$ as the source messages generated by the source node $\sigma_i$ for $i = 1, 2, \ldots, s$ when using the linear network code $C$ for the model $(N,f,b)$. By the admissibility of the code $C$ (i.e., the computability condition is satisfied for $C$), we can apply the decoding function $\varphi$ of $C$ on $y_\rho$ to compute with zero error
\[
\sum_{i=1}^s x_i \cdot B^{-1} = \left( \sum_{i=1}^s x_i \right) \cdot B^{-1}.
\]

Then, the admissibility of the linear network code $C$ implies that there exists an $[\text{In}(\rho)] \times R$ matrix $D$ in $\mathbb{F}_q$ such that
\[
G_\rho \cdot D = \begin{bmatrix} I_R \\
I_R \\
\vdots \\
I_R \end{bmatrix}_{R \times R}
\]
\[
(54)
\]
where $I_R$ stands for the $R \times R$ identity matrix. Immediately, we have
\[
y_\rho \cdot D = (x_1 B^{-1} x_2 B^{-1} \cdots x_s B^{-1}) \cdot G_\rho \cdot D
\]
\[
(53)
\]

3 Here, $\mathbf{O}$ stands for an all-zero $R \times R$ matrix. In the following, we always use $\mathbf{O}$ to denote an all-zero square matrix whose size is clear from the context.
\[ = \sum_{i=1}^{s} x_i \cdot B^{-1} = \left( \sum_{i=1}^{s} x_i \right) \cdot B^{-1}. \]

By multiplying both sides of (53) by \( B \), we obtain \( \sum_{i=1}^{s} x_i \), or equivalently, \( \left( \sum_{i=1}^{s} m_i \right) \). Thus, the \( R - r \) function values
\[ \sum_{i=1}^{s} m_i = \left( \sum_{i=1}^{s} m_{i,j}, j = 1, 2, \cdots, R - r \right) \]
are computed at \( \rho \) with zero error. We thus have verified the computability condition of the code \( C \), and at the same time specified the decoding function \( \hat{\varphi} \), which indeed is linear.

**Verification of the Security Condition:** We let \( X_i = (M_i, K_i), i = 1, 2, \cdots, s \) and \( X_S = (X_1 X_2 \cdots X_s) \). For each \( e \in \mathcal{E} \), we let \( Y_e = X_S \cdot \hat{h}_e \), which is the random message transmitted on the edge \( e \). We consider an arbitrary wiretap set \( W \in \mathcal{W} \), and will prove that
\[ H(M_S \mid Y_W) = H(M_S). \]

Toward this end, it suffices to prove that
\[ \Pr(M_S = m_S \mid Y_W = y_W) = \Pr(M_S = m_S) \quad (55) \]
for any vector of source messages \( m_S = (m_1, m_2, \cdots, m_s) \in F_q^{|R-r|s} \) and any \( y_W \triangleq (y_e : e \in W) \in F_q^{|W|} \) such that \( \Pr(Y_W = y_W) > 0 \).

First, we readily see that
\[ \Pr(M_S = m_S) = \frac{1}{q^{|R-r|s}}, \forall m_S \in F_q^{|R-r|s}. \quad (56) \]

Next, we consider
\[ \Pr(M_S = m_S \mid Y_W = y_W) = \frac{\Pr(M_S = m_S, Y_W = y_W)}{\Pr(Y_W = y_W)} \]
\[ = \frac{\sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S, Y_W = y_W)}{\sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S, Y_W = y_W)}, \quad (57) \]
where in (57), \( \hat{m}_S = (\hat{m}_1 \hat{m}_2 \cdots \hat{m}_s) \in F_q^{|R-r|s} \) is a row vector with \( \hat{m}_i \) being a row \((R - r)\)-vector for \( i = 1, 2, \cdots, s \) and similarly, \( \hat{k}_S = (\hat{k}_1 \hat{k}_2 \cdots \hat{k}_s) \in F_q^{|R-r|s} \) is a row vector with \( \hat{k}_i \) being a row \( r \)-vector for \( i = 1, 2, \cdots, s \).

In the following, we calculate the numerator and denominator of (57), respectively.

**For the numerator of (57),** we have
\[ \sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S, Y_W = y_W) \]
\[ = \sum_{k_S \in F_q^{|R-r|s}} \Pr(Y_W = y_W \mid M_S = m_S, K_S = k_S) \cdot \Pr(M_S = m_S, K_S = k_S) \]
\[ = \sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S), \quad (58) \]

where, in (58), we let
\[ H_W = [\hat{h}_e : e \in W]. \]

**For the denominator of (57),** we have
\[ \sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S, Y_W = y_W) \]
\[ = \sum_{k_S \in F_q^{|R-r|s}} \Pr(Y_W = y_W \mid M_S = \hat{m}_S, K_S = \hat{k}_S) \cdot \Pr(M_S = \hat{m}_S, K_S = \hat{k}_S) \]
\[ = \sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = \hat{m}_S, K_S = \hat{k}_S), \quad (59) \]
where the equality (59) follows from
\[ \Pr(Y_W = y_W \mid M_S = \hat{m}_S, K_S = \hat{k}_S) = \begin{cases} 1, & \text{if } (\hat{m}_S \hat{k}_S) \cdot H_W = y_W; \\ 0, & \text{otherwise}. \end{cases} \]

By substituting (58) and (59) into (57), we obtain
\[ \Pr(M_S = m_S \mid Y_W = y_W) = \frac{\sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S, Y_W = y_W)}{\sum_{k_S \in F_q^{|R-r|s}} \Pr(M_S = m_S, K_S = k_S)} \]
\[ = \frac{\# \{ \hat{k}_S \in F_q^{|R-r|s} : (\hat{m}_S \hat{k}_S) \cdot H_W = y_W \}}{\# \{ \hat{m}_S \hat{k}_S \in F_q^{|R-r|s} \times F_q^{|R-r|s} : (\hat{m}_S \hat{k}_S) \cdot H_W = y_W \}}, \quad (60) \]
where, in (60), we use “\#\{\}” to denote the cardinality of the set.

Further, we let \( \text{Rank}(H_W) = r' \). Clearly, \( r' \leq |W| \leq r \). For the denominator of (60), it is easy to calculate that
\[ \# \{ \hat{m}_S \hat{k}_S \in F_q^{|R-r|s} \times F_q^{|R-r|s} : (\hat{m}_S \hat{k}_S) \cdot H_W = y_W \} = q^{|R-r|s}. \]

Next, we focus on the numerator of (60), where we note that
\[ (m_S \hat{k}_S) = (m_1 k_1 m_2 k_2 \cdots m_s k_s). \]
We denote by \( I_{R \times R}^{(r-r)} \) the submatrix of the \( R \times R \) identity matrix \( I_R \) consisting of its first \( R - r \) columns. We further define the \( Rs \times (R-r)s \) matrix
\[ \Gamma = [I_R^{(r-r)} \quad I_R^{(R-r)} \cdots I_R^{(R-r)}]_{s \times s \text{ diagonal}} \]
\[ = \begin{bmatrix} I_R^{(r-r)} & 0 & \cdots & 0 \\ 0 & I_R^{(R-r)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_R^{(R-r)} \end{bmatrix}_{Rs \times (R-r)s}, \]
where \( 0 \) stands for the \( R \times (R-r) \) zero matrix. Further, we can see that
\[ \# \{ \hat{k}_S \in F_q^{|R-r|s} : (m_S \hat{k}_S) \cdot H_W = y_W \}
\[ = \# \{ \hat{m}_S \hat{k}_S \in F_q^{|R-r|s} \times F_q^{|R-r|s} : \]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\[
(\tilde{m}_S \tilde{k}_S) \cdot \begin{bmatrix} H_W & \Gamma \end{bmatrix} = (y_W \cdot m_S).
\] (62)

It thus suffices to calculate the number of solutions \((\tilde{m}_S \tilde{k}_S)\) of the equation
\[
(\tilde{m}_S \tilde{k}_S) \cdot \begin{bmatrix} H_W & \Gamma \end{bmatrix} = (y_W \cdot m_S).
\]

Toward this end, we write
\[
\begin{bmatrix} H_W & \Gamma \end{bmatrix} = [\mathbf{\tilde{B}}^{-1} \cdot G_W \mathbf{\tilde{B}}^{-1} \cdot \mathbf{\tilde{B}} \cdot \Gamma],
\] (63)

where \(G_W = [\tilde{g}_e : e \in W]\). We let \(B^{(R-r)}\) be the \(R \times (R-r)\) submatrix of \(\mathbf{\tilde{B}}\) consisting of its first \(R-r\) columns, i.e.,
\[
B^{(R-r)} = [\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_{R-r}].
\]

Then, we see that
\[
\mathbf{\tilde{B}} \cdot \Gamma = [B^{(R-r)} B^{(R-r)} \ldots B^{(R-r)}]^\text{diagonal}_s.
\]

and so
\[
\begin{bmatrix} G_W & \mathbf{\tilde{B}} \cdot \Gamma \end{bmatrix}
= \begin{bmatrix} \tilde{g}_e : e \in W \end{bmatrix} \begin{bmatrix} B^{(R-r)} \ B^{(R-r)} \ldots \ B^{(R-r)} \end{bmatrix}^\text{diagonal}_s
= \begin{bmatrix} \tilde{g}_{e^{(\sigma_1)}} : e \in W & B^{(R-r)} & 0 & \ldots & 0 \\
\tilde{g}_{e^{(\sigma_2)}} : e \in W & 0 & B^{(R-r)} & \ldots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\tilde{g}_{e^{(\sigma_s)}} : e \in W & 0 & \ldots & \ldots & B^{(R-r)} \end{bmatrix}.
\] (64)

Further, we recall that the \(R-r\) vectors \(\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_{R-r}\) are linearly independent and for each wiretap set \(W \in \mathcal{W}_r\), we have
\[
\langle \tilde{b}_j : 1 \leq j \leq R-r \rangle \cap \langle \tilde{g}_e^{(\sigma)} : e \in W \rangle = \{\tilde{0}\},
\]
\[
\forall i = 1, 2, \ldots, s
\]
(cf. (49)). Together with (63), (64) and the invertibility of \(\mathbf{\tilde{B}}\), this implies that
\[
\text{Rank} \left[ \begin{bmatrix} H_W & \Gamma \end{bmatrix} \right] = \text{Rank} \left[ \begin{bmatrix} G_W & \mathbf{\tilde{B}} \cdot \Gamma \end{bmatrix} \right]
= \text{Rank}(G_W) + \text{Rank}(\mathbf{\tilde{B}} \cdot \Gamma)
= \text{Rank}(H_W) + s \cdot \text{Rank}(B^{(R-r)})
= r' + (R-r)s.
\]

We thus obtain that
\[
\# \left\{ (\tilde{m}_S \tilde{k}_S) \in F_q^{(R-r)s} \times F_q^{rs} : (\tilde{m}_S \tilde{k}_S) \cdot \begin{bmatrix} H_W & \Gamma \end{bmatrix} = (y_W \cdot m_S) \right\}
= q^{rs} - \text{Rank} \left[ \begin{bmatrix} H_W & \Gamma \end{bmatrix} \right] = q^{r-s-r'},
\]
and also by (62),
\[
\# \left\{ k_S \in F_q^{rs} : (m_S \tilde{k}_S) \cdot H_W = y_W \right\} = q^{r-s-r'}.
\] (65)

Combining (61) and (65) with (60), we immediately obtain that
\[
\text{Pr}(M_S = m_S | Y_W = y_W) = \frac{1}{q^{(R-r)s}}.
\] (66)

Comparing (56) and (66), we have proved the equality (55) and thus verified the security condition.

### E. The Required Field Size of the Code Construction

In this subsection, we present upper bounds on the minimum required field size of the existence of an admissible \((R-r,1)\) linear secure network code \(\mathcal{C}\) for the secure model \((N,f,r)\) with \(0 \leq r \leq R \leq C_{\text{min}}\). Following the upper bounds, we further show that by our code construction, for the algebraic sum \(f\) over any finite field \(F_q\), we can always construct a \(F_q\)-valued admissible (vector-) linear secure network code of rate up to \(C_{\text{min}} - r\) for the model \((N,f,r)\) with security level \(0 \leq r \leq C_{\text{min}}\).

By applying the notion of primary minimum cut, we first prove in the next theorem a non-trivial upper bound on the minimum required field size of the existence of an admissible linear secure network code. This upper bound, which is graph-theoretic, only depends on the network topology and the required security level. Before presenting the upper bound, we recall that for an edge subset \(W, \tilde{W}\) denotes the primary minimum cut separating \(W\) from \(D_W\) and \(W = \tilde{W}\) is the condition for \(W\) to be primary.

**Theorem 13:** Consider the model of secure network function computation \((N,f,r)\), where the target function \(f\) is the algebraic sum over a finite field \(F_q\) and the security level \(r\) satisfies \(0 \leq r \leq C_{\text{min}}\). Let
\[
\mathcal{W}_r = \{ W \in \mathcal{W}_r : W = \tilde{W} \text{ with } |W| = r \}.
\]

Then, for any nonnegative integer \(r\) with \(r \leq R \leq C_{\text{min}}\), there exists an \(F_q\)-valued admissible \((R-r,1)\) linear secure network code for \((N,f,r)\) if the field size \(q\) satisfies
\[
q > s \cdot |\mathcal{W}_r|.
\] (67)

**Proof:** Consider an arbitrary admissible secure network code \(\mathcal{C}\) for the model \((N,f,\mathcal{W}_r)\). We first claim that the code \(\mathcal{C}\) is also admissible for the model \((N,f,r)\). It is evident that \(\mathcal{C}\) satisfies the computability condition for \((N,f,r)\). Then, it suffices to prove that \(\mathcal{C}\) satisfies the security condition for \((N,f,r)\), i.e., \(I(Y_W;M_S) = 0, \forall W \in \mathcal{W}_r\) provided that \(I(Y_W;M_S) = 0, \forall W \in \mathcal{W}_r\). Toward this end, we consider an arbitrary wiretap set \(W \in \mathcal{W}_r\) and let \(W'\) be a cut separating \(W\) from \(D_W\) with \(|W'| = \text{mincut}(D_W, W') = r\), where \(\text{mincut}(D_W, W')\) denotes the minimum cut capacity separating \(W'\) from \(D_W\). Further, we consider the primary minimum cut \(\tilde{W}'\) separating \(W'\) from \(D_W\). By Lemma 5, \(\tilde{W}'\) is primary, or equivalently, \(\tilde{W} = \tilde{W}'\). Together with
\[
|\tilde{W}'| = \text{mincut}(D_W, W') = |W'| = r,
\]
we have \(\tilde{W} \in \mathcal{W}_r\) and thus \(I(Y_{\tilde{W}};M_S) = 0\) by the security condition of the code \(\mathcal{C}\) for \((N,f,\mathcal{W}_r)\). This further implies \(I(Y_W;M_S) = 0\), because \(Y_W\) is a function of \(Y_{W'}\) and \(Y_{W'}\) is a function of \(Y_{\tilde{W}}\) by the mechanism of network coding.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Next, we will consider the field size $q$ for constructing such an $\mathbb{F}_q$-valued admissible $(R-r, 1)$ linear secure network code for $(\mathcal{N}, f, \mathcal{W}_{\sigma}^*)$ by applying the code construction in Section IV-C. First, let $\mathcal{C}$ be an $\mathbb{F}_q$-valued admissible $(R, 1)$ linear network code for the model $(\mathcal{N}, f)$ of which all the global encoding vectors are

$$\tilde{g}_e = \begin{bmatrix} \tilde{g}^{(i_1)}_e \\ \vdots \\ \tilde{g}^{(i_s)}_e \end{bmatrix}, \quad e \in \mathcal{E}.$$ 

Similar to the verification of the security condition in Section IV-D, we readily see that an $\mathbb{F}_q$-valued admissible $(R-r, 1)$ linear secure network code for the secure model $(\mathcal{N}, f, \mathcal{W}_{\sigma}^*)$ can be constructed if there exist $R$ linearly independent $\mathbb{F}_q$-valued column $R$-vectors $\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_R$ such that the first $R-r$ vectors $\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{R-r}$ satisfy the condition that for all the wiretap sets $W \subseteq \mathcal{W}_{\sigma}^*$,

$$\langle \tilde{b}_j : 1 \leq j \leq R-r \rangle \cap \langle \tilde{g}^{(i)}_e : e \in W \rangle = \{0\},$$

$$\forall i = 1, 2, \cdots, s. \tag{68}$$

In the following, we will prove that such $R$ vectors $\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_R$ exist if the inequality (67) holds. The techniques involved here are standard in the literature.

For notational simplicity, we let

$$B_j = \langle \tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_{j-1} \rangle, \quad \forall 1 \leq j \leq R,$$

and

$$\mathcal{L}^{(i)}_W = \langle \tilde{g}^{(i)}_e : e \in W \rangle, \quad \forall 1 \leq i \leq s \quad \text{and} \quad W \subseteq \mathcal{W}_{\sigma}^*.$$ 

Now, we choose the $R$ vectors $\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_R$ in $\mathbb{F}_q^R$ sequentially as follows:

- For $1 \leq j \leq R-r$, we choose

$$\tilde{b}_j \in \mathbb{F}_q^R \setminus \bigcup_{W \subseteq \mathcal{W}_{\sigma}^*} \bigcup_{i=1}^s (\mathcal{L}^{(i)}_W + B_{j-1}); \tag{69}$$

- For $R-r + 1 \leq j \leq R$, we choose

$$\tilde{b}_j \in \mathbb{F}_q^R \setminus B_{j-1}. \tag{70}$$

By (69) and (70), we have

$$\tilde{b}_1 \neq \tilde{0} \quad \text{and} \quad \tilde{b}_j \in \mathbb{F}_q^R \setminus B_{j-1}, \quad 2 \leq j \leq R,$$

which implies the linear independence of $\tilde{b}_1, \tilde{b}_2, \cdots, \tilde{b}_R$. On the other hand, it follows from (69) that the condition (68) is satisfied for each wiretap set $W \subseteq \mathcal{W}_{\sigma}^*$.

It now remains to prove that all the sets on the right hand sides of (69) and (70) are nonempty provided that $q > s \cdot |\mathcal{W}_{\sigma}^*|$. For $1 \leq j \leq R-r$, by (69), we obtain that

$$\left| \mathbb{F}_q^R \setminus \bigcup_{W \subseteq \mathcal{W}_{\sigma}^*} \bigcup_{i=1}^s (\mathcal{L}^{(i)}_W + B_{j-1}) \right| \geq q^R - \sum_{W \subseteq \mathcal{W}_{\sigma}^*} \sum_{i=1}^s |\mathcal{L}^{(i)}_W + B_{j-1}|$$

where the inequality (71) holds because for any $1 \leq i \leq s, 1 \leq j \leq R-r$ and $W \subseteq \mathcal{W}_{\sigma}^*$, we have

$$\dim(\mathcal{L}^{(i)}_W) \leq r \quad \text{and} \quad \dim(B_{j-1}) \leq R-r-1,$$

implying that

$$\dim(\mathcal{L}^{(i)}_W + B_{j-1}) \leq \dim(\mathcal{L}^{(i)}_W) + \dim(B_{j-1}) \leq r + R-r-1 = R-1.$$ 

For $R-r+1 \leq j \leq R$, by (70) we have

$$|\mathbb{F}_q^R \setminus B_{j-1}| = q^R - q^{j-1} \geq q^R - q^{R-1} > 0.$$

The theorem is thus proved.

We can readily see that

$$|\mathcal{W}_{\sigma}^*| \leq \left| \{ W \subseteq \mathcal{E} : |W| = r \} \right| = \left( \frac{|\mathcal{E}|}{r} \right).$$

Following Theorem 13, this immediately gives an upper bound on the required field size in closed form that does not depend on the network topology.

**Corollary 14:** Consider the model of secure network function computation $(\mathcal{N}, f, r)$, where the target function $f$ is the algebraic sum over a finite field $\mathbb{F}_q$ and the security level $r$ satisfies $0 \leq r \leq C_{\min}$. Then, for any nonnegative integer $R$ with $r \leq R \leq C_{\min}$, there exists an $\mathbb{F}_q$-valued admissible $(R-r, 1)$ linear secure network code for $(\mathcal{N}, f, r)$ if the field size $q$ satisfies $q > s \cdot \left( \frac{|\mathcal{E}|}{r} \right)$.

**Remark 4:** For the case that the security level $r = 0$, i.e., to compute the algebraic sum $f$ over the network $\mathcal{N}$ without security constraint, we have $|\mathcal{W}_{\sigma}^*| = 1$ since $\mathcal{W}_0 = \mathcal{W}_{\sigma}^* = \{\{0\}\}$. On the other hand, $\left( \frac{|\mathcal{E}|}{0} \right) = 1$. Thus, the upper bounds in Theorem 13 and Corollary 14 are both equal to $s$, the number of source nodes.

We continue to consider the secure model $(\mathcal{N}, f, r)$, where the target function $f$ is the algebraic sum on a finite field $\mathbb{F}_q$ and the security level $r \leq C_{\min}$. For the case that the field size $q > s \cdot |\mathcal{W}_{\sigma}^*|$, by Theorem 13, we can construct an $\mathbb{F}_q$-valued admissible $(R-r, 1)$ linear secure network code for any $1 \leq R \leq C_{\min}$ by using our code construction. Next, we consider the case that the field size $q \leq s \cdot |\mathcal{W}_{\sigma}^*|$. We take an extension field $\mathbb{F}_{q^L}$ of $\mathbb{F}_q$ such that the size of the extension field $q^L > s \cdot |\mathcal{W}_{\sigma}^*|$. Then, an $\mathbb{F}_{q^L}$-valued admissible $(R-r, 1)$ linear secure network code can be obtained by our code construction. Note that the extension field $\mathbb{F}_{q^L}$ can be viewed as an $L$-dimensional vector space over $\mathbb{F}_q$ with the basis $\{1, \alpha, \alpha^2, \cdots, \alpha^{L-1}\}$, where $\alpha$ is a primitive element of $\mathbb{F}_{q^L}$. Hence, the $\mathbb{F}_{q^L}$-valued $(R-r, 1)$ linear secure network code for computing the algebraic sum on $\mathbb{F}_{q^L}$ can be regarded as an $(\langle (R-r)L, L \rangle)$ linear secure network code for computing the algebraic sum $f$ on $\mathbb{F}_q$, which also has the same secure computing rate $R-r$. Combining the two cases, for the algebraic sum $f$ over any finite field $\mathbb{F}_q$, we can always construct an $\mathbb{F}_q$-valued admissible (vector-) linear secure network code of rate up to $C_{\min} - r$ for the model $(\mathcal{N}, f, r)$ with security level $0 \leq r \leq C_{\min}$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
\( \sigma_1 \) and \( \sigma_2 \) sequentially generate 4 symbols in \( \mathbb{F}_2 \), say, \( m_{i,1}, m_{i,2}, m_{i,3}, m_{i,4} \), which are regarded as 2 elements in \( \mathbb{F}_4 \):

\[
x_{i,1} \triangleq m_{i,1} + m_{i,2} \cdot \alpha \quad \text{and} \quad x_{i,2} \triangleq m_{i,3} + m_{i,4} \cdot \alpha.
\]

Using the code \( C \), the sink node \( \rho \) receives

\[
y_{e_8} = (x_{1,1} \cdot x_2 + x_2 \cdot x_{2,1}) \cdot \bar{g}_{e_8} = x_{1,1} + x_{2,1} = (m_{1,1} + m_{2,1}) + (m_{1,2} + m_{2,2}) \cdot \alpha,
\]

\[
y_{e_9} = (x_{1,1} \cdot x_2 + x_2 \cdot x_{2,1}) \cdot \bar{g}_{e_9} = x_{1,2} + x_{2,2} = (m_{1,3} + m_{2,3}) + (m_{1,4} + m_{2,4}) \cdot \alpha.
\]

Thus, the 4 function values \( f(m_{1,j}, m_{2,j}) = m_{1,j} + m_{2,j}, j = 1, 2, 3, 4 \) are computed with zero error at the sink node \( \rho \).

Based on the code \( C \), we will construct an \( \mathbb{F}_4 \)-valued (1, 1) linear secure network code \( C \) on the network \( \mathcal{N} \) for securely computing the algebraic sum over \( \mathbb{F}_4 \) with the security level \( r = 1 \). This code \( C \) can also be similarly regarded as an \( \mathbb{F}_2 \)-valued (2, 2) linear secure network code for \( \langle N, f, r \rangle \).

By construction, it suffices to construct an \( \mathbb{F}_2 \)-valued 2 \( \times \) 2 invertible matrix \( B = [\bar{b}_{1} \quad \bar{b}_{2}] \) such that for any edge \( e \in E \), or equivalently, for any wiretap set \( \{e\} \in \mathcal{W}_1 \),

\[
\langle \bar{b}_1 \rangle \cap \langle \bar{g}_{e_8}^{(\sigma_i)} \rangle = \{0\}, \quad \forall i = 1, 2,
\]

(cf. (49)). To satisfy the above condition, we choose

\[
\bar{b}_1 = \begin{bmatrix} 1 \\ \alpha \end{bmatrix} \quad \text{and} \quad \bar{b}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},
\]

i.e.,

\[
\bar{B} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}.
\]

Then, \( B^{-1} = [1 \quad 0] \begin{bmatrix} \alpha & 1 \\ 0 & 1 \end{bmatrix} \). We further let

\[
\bar{B} = \begin{bmatrix} B \\ \begin{bmatrix} B \\ 0 \end{bmatrix} \end{bmatrix}_{\text{diagonal}} = \begin{bmatrix} B \\ 0 \end{bmatrix},
\]

and then \( \bar{B}^{-1} \triangleq \begin{bmatrix} B^{-1} \\ B^{-1} \end{bmatrix}_{\text{diagonal}} = \bar{B} \). We thus have obtained an \( \mathbb{F}_2 \)-valued admissible (1, 1) linear secure network code \( C \) on the network \( \mathcal{N} \) for securely computing the algebraic sum over \( \mathbb{F}_2 \) with security level \( r = 1 \). The global encoding vectors of \( \bar{C} \) are \( \bar{h}_e = \bar{B}^{-1} \cdot \bar{g}_e, e \in E, \) i.e.,

\[
\bar{h}_{e_1} = \begin{bmatrix} 1 + \alpha \\ 0 \\ 0 \end{bmatrix}, \quad \bar{h}_{e_2} = \begin{bmatrix} 0 \\ 1 + \alpha \\ 0 \end{bmatrix}, \quad \bar{h}_{e_3} = \begin{bmatrix} 0 \\ 0 \\ 1 + \alpha \end{bmatrix}, \quad \bar{h}_{e_4} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, 
\]

\[
\bar{h}_{e_5} = \bar{h}_{e_6} = \bar{h}_{e_7} = \bar{h}_{e_8} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \bar{h}_{e_9} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \bar{h}_{e_{10}} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.
\]

In using the obtained code \( \bar{C} \), for \( i = 1, 2 \), let \( m_{i,1} \) and \( m_{i,2} \) in \( \mathbb{F}_2 \) be two symbols generated by the source node \( \sigma_i \), which can be regarded as the element \( m_i \triangleq m_{i,1} + m_{i,2} \cdot \alpha \) in \( \mathbb{F}_4 \). Further, let \( k_1 \in \mathbb{F}_4 \) and \( k_2 \in \mathbb{F}_4 \) be two arbitrary outputs of the random keys \( K_1 \) and \( K_2 \), respectively, where \( K_1 \) and \( K_2 \) are two i.i.d. random variables with the uniform distribution on \( \mathbb{F}_4 \). Then, we let \( x_1 = (m_1, k_1) \), \( x_2 = (m_2, k_2) \), and \( x_3 = (x_1, x_2) = (m_1, k_1, m_2, k_2) \).

We use \( y_{e_c} \), which takes values in \( \mathbb{F}_2 \), to denote the message transmitted on each edge \( e \in E \). By the global encoding
vectors of \( \hat{C} \) (cf. \eqref{eq:72}), the messages \( y_e = x_S \cdot \tilde{h}_e \) transmitted on the edges \( e \in E \) are

\[
\begin{align*}
    y_{e_1} &= m_1 + (1 + \alpha) \cdot k_1, \\
    y_{e_2} &= k_1, \\
    y_{e_3} &= m_2 + \alpha \cdot k_2, \\
    y_{e_4} &= m_2 + (1 + \alpha) \cdot k_2, \\
    y_{e_5} &= y_{e_7} = k_1 + m_2 + \alpha \cdot k_2, \\
    y_{e_8} &= m_1 + \alpha \cdot k_1 + m_2 + \alpha \cdot k_2, \\
    y_{e_9} &= k_1 + k_2. 
\end{align*}
\]

We can readily verify the computability and security conditions for the code \( \hat{C} \). More precisely, by the two messages \( y_{e_8} \) and \( y_{e_9} \) received at the sink node \( \rho \), the algebraic sum \( m_1 + m_2 \) over \( \mathbb{F}_4 \), or equivalently, the algebraic sums \( m_{1,1} + m_{2,1} \) and \( m_{1,2} + m_{2,2} \) over \( \mathbb{F}_2 \), are computed with zero error at the sink node \( \rho \). On the other hand, it is also easy to check that the wiretapper cannot obtain any information about the source messages \( m_1 \) and \( m_2 \) when any one edge is eavesdropped.

For the secure model \( (\mathcal{N}, f, r) \) as discussed in the above example, by Theorem 13, we can construct an \( \mathbb{F}_q \)-valued admissible \((1,1)\) linear secure network code for computing the algebraic sum over \( \mathbb{F}_q \) on the network \( \mathcal{N} \) with security level \( r = 1 \) if the field size \( q > 2 \cdot |\mathcal{V}_r^1| = 14 \), where \( \mathcal{W}_r^1 = \{e_1, e_2, e_3, e_4, e_5, e_8, e_9\} \). However, we see in the example that the finite field \( \mathbb{F}_4 \) is sufficient for our code construction. This implies that for our code construction, the bound \eqref{eq:67} in Theorem 13 on the field size is only sufficient but far from being necessary.

We continue to consider the secure model \( (\mathcal{N}, f, r) \) in the above example. In Fig. 5, we show an \( \mathbb{F}_2 \)-valued admissible \((1,1)\) linear secure network code which is also an optimal code achieving the secure computing capacity \( \hat{C}(\mathcal{N}, f, r) = 1 \). Note that this code requires only \( \mathbb{F}_2 \), which is smaller than \( \mathbb{F}_4 \) required in the example. However, this code is yet to be obtained by our code construction. More precisely, according to the code construction in Section IV-C, we construct an admissible linear secure network code for the secure model \( (\mathcal{N}, f, r) \) by using a common linear transformation at all the source nodes on an admissible linear network code for \( (\mathcal{N}, f) \).

As such, the algebraic sum of all the keys \( \sum_{i=1}^{s} k_i \) can be always computed with zero error at the sink node \( \rho \) (cf. IV-D), e.g., \( k_1 + k_2 \) in the example (cf. \eqref{eq:73}). However, we can readily see that in Fig. 5, the code is not the case, where the key \( k_1 \) of the source node \( \sigma_1 \) is cancelled out before arriving to \( \rho \).

V. CONCLUSION

Secure network function computation has been put forward in this paper. As the first work on this problem, we have investigated the special case of securely computing a linear function over a finite field with a wiretapper who can eavesdrop any subset of edges up to a certain size but is not allowed to obtain any information about the source messages. We have proved an upper bound on the secure computing capacity, which is applicable to arbitrary network topologies and arbitrary security levels. Since this upper bound is not in closed form, we also have proposed an efficient graph-theoretic approach to compute it in a linear time of the number of edges in the network. Furthermore, we have developed a code construction for linear function-computing secure network codes. Based on this construction, we have obtained a lower bound on the secure computing capacity and provided two sufficient conditions in terms of the network topology on the tightness of the lower bound. By combining the obtained upper and lower bounds, we have fully characterized the secure computing capacities for some classes of secure models. In fact, the obtained upper bound is tight for all the specific models in the examples of the current paper. However, this upper bound is not always tight. A counterexample can be found in a recent work [31]. Therein, an upper bound on the secure computing capacity was also claimed but in general this upper bound is invalid. A corrected upper bound has been obtained by the same authors of [31] (M. Xu, personal communication, June 17, 2023).

For the model considered in the current paper, several interesting problems still remain open, such as whether other code constructions can be designed for smaller field sizes or to achieve higher secure computing rates, what the minimum size of the random key or the minimum entropy of the random key for each source node is, etc. In the general setup of the model of secure network function computation, we can also consider different criteria for security. In Part II of this paper, we will investigate securely computing a linear function over a finite field, where the wiretapper is not allowed to obtain any information about this linear function.

Another interesting problem is the possible connection between secure network function computation discussed in this paper and secure multiparty computation which has been well studied in the literature, e.g., [32], [33], [34], [35], [36] and [37].

APPENDIX A

CONSTRUCTION OF LINEAR NETWORK CODES FOR THE MODEL \( (\mathcal{N}, f) \)

We refer the reader to the definition of an admissible \((R, 1)\) linear network code for the model \( (\mathcal{N}, f) \) in Section IV-C, where some notations therein are adopted in this appendix.
Consider the single-source multicast problem on the reversed network $N^\top \triangleq (G^\top, S, \rho)$ of $N$, where the node $\rho$, regarded as the single source node, is required to multicast the source message on the reversed graph $G^\top$ to the nodes $\sigma_i \in S$, regarded as all the sink nodes. On $G^\top$, the tail and head node of an edge $e \in E$ are denoted by tail$^G(e)$ and head$^G(e)$, respectively. Clearly, we have tail$^G(e) = \text{head}(e)$ and head$^G(e) = \text{tail}(e)$. For a node $v$ in $G^\top$, we let $\text{In}^{G^\top}(v)$ and $\text{Out}^{G^\top}(v)$ be the set of input edges and the set of output edges on the reversed graph $G^\top$, respectively. Then, we have

$$\text{In}^{G^\top}(v) = \text{Out}(v) \quad \text{and} \quad \text{Out}^{G^\top}(v) = \text{In}(v). \quad (74)$$

For a node $\sigma_i \in S$, we see that

$$\mincut^{G^\top}(\rho, \sigma_i) = \mincut(\sigma_i, \rho),$$

where $\mincut^{G^\top}(\rho, \sigma_i)$ denotes the minimum cut capacity separating $\sigma_i$ from $\rho$ on $G^\top$. Thus,

$$\min_{1 \leq i \leq s} \mincut^{G^\top}(\rho, \sigma_i) = \min_{1 \leq i \leq s} \mincut(\sigma_i, \rho),$$

which are both denoted by $C_{\min}$.

We now consider an $\mathbb{F}_q$-valued rate-$R$ (scalar) linear network code for the single-source multicast problem on the reversed network $N^\top$, where $\mathbb{F}_q$ is a finite field of order $q$ and the rate $R$ is a nonnegative integer not larger than $C_{\min}$, the theoretical maximum rate at which $\mathbb{F}_q$ can multicast the source message to all the sink nodes in $S$ on $N^\top$. For the rate $R$, we introduce $R$ imaginary source edges connecting to $\rho$, denoted by $d_1', d_2', \ldots, d_R'$, respectively, and let $\text{In}^G(\rho) = \{d_1', d_2', \ldots, d_R'\}$. For the source message $x = (x_1 x_2 \cdots x_R) \in \mathbb{F}_q^R$ of $R$ source symbols generated by $\rho$, we assume without loss of generality that $x_i$ is transmitted on the $i$th imaginary channel $d_i'$, $1 \leq i \leq R$.

**Definition 1:** For the single-source multicast problem on the reversed network $N^\top \triangleq (G^\top, S, \rho)$, an $\mathbb{F}_q$-valued rate-$R$ linear network code $C_{N^\top}$ consists of a local encoding kernel $K_v$ for each non-sink node $v$ in $V \setminus S$ and a decoding matrix $F_{\sigma_i}$ for each sink node $\sigma_i \in S$, as specified below:

- For each non-sink node $v \in V \setminus S$, the local encoding kernel is an $\mathbb{F}_q$-valued $|\text{In}^{G^\top}(v)| \times |\text{Out}^{G^\top}(v)|$ matrix

$$K_v = [k_{d,e} : d \in \text{In}^{G^\top}(v), e \in \text{Out}^{G^\top}(v)],$$

where $k_{d,e} \in \mathbb{F}_q$ is called the local encoding coefficient for the adjacent edge pair $(d, e)$;

- For each sink node $\sigma_i \in S$, the decoding matrix is an $\mathbb{F}_q$-valued $R \times |\text{In}^{G^\top}(\sigma_i)|$ matrix

$$F_{\sigma_i} = [f_e : e \in \text{In}^{G^\top}(\sigma_i)]$$

induced by the local encoding kernels $K_v$,

where for each edge $e \in E$, $f_e$ is a column $R$-vector, called the global encoding kernel of $e$, which can be calculated recursively according to the reverse order of “$\prec$” (cf. the second paragraph of Section IV-C for the order “$\prec$”) by

$$f_e = \sum_{d \in \text{In}^{G^\top}(\text{tail}^{G^\top}(e))} k_{d,e} \cdot f_d \quad (75)$$

with the boundary condition that $f_{\text{In}}$, $1 \leq i \leq R$ form the standard basis of the vector space $\mathbb{F}_q^R$.

We use $y_e$ to denote the symbol transmitted on $e$, $\forall e \in \text{In}^{G^\top}(\rho) \cup E$. With $y_{d_i'} = x_i = x \cdot f_{d_i'}, 1 \leq i \leq R$, each $y_e$ for $e \in E$ can be calculated recursively according to the reverse order of “$\prec$” on the edges in $E$ by

$$y_e = \sum_{d \in \text{In}^{G^\top}(v)} k_{d,e} \cdot y_d,$$

where $v = \text{tail}^{G^\top}(e)$, and together with (75), we have

$$y_e = x \cdot f_e, \quad \forall e \in E.$$

For each sink node $\sigma_i \in S$, we further have

$$(y_e : e \in \text{In}^{G^\top}(\sigma_i)) = x \cdot [f_e : e \in \text{In}^{G^\top}(\sigma_i)] = x \cdot F_{\sigma_i}.$$  

Then, the source message $x$ can be decoded at $\sigma_i$ if and only if $\text{Rank}(F_{\sigma_i}) = R$. Accordingly, the rate-$R$ code $C_{N^\top}$ is called decodable if $\text{Rank}(F_{\sigma_i}) = R$, $\forall \sigma_i \in S$, or equivalently, for each decoding matrix $F_{\sigma_i}$, there exists an $\mathbb{F}_q$-valued $|\text{In}^{G^\top}(\sigma_i)| \times R$ matrix $K_{\sigma_i}$ such that

$$F_{\sigma_i} \cdot K_{\sigma_i} = I_R, \quad (76)$$

where $I_R$ stands for the $R \times R$ identity matrix.

Furthermore, we set $k_{d,e} = 0$ for all non-adjacent edge pairs $(d, e)$ on $G^\top$ which satisfy $\text{head}^{G^\top}(d) \neq \text{tail}^{G^\top}(e)$. We now define an $|E| \times |E|$ matrix

$$K = (k_{d,e})_{d \in E, e \in E}. $$

We can see that according to the reverse order of “$\prec$”, $K$ is lower triangular and all the diagonal elements are equal to zero. For an edge subset $\eta \subseteq E$ on $G^\top$, we let

$$I_\eta = [I_e : e \in \eta, \quad 0 : e \in E \setminus \eta]$$

be an $|\eta| \times |E|$ matrix in which the rows are indexed by the edges in $\eta$ and the columns are indexed by the edges in $E$, where $I_e$ is the column $|\eta|$-vector whose component indexed by the edge $e$ is equal to 1 while all other components are equal to 0, and $0$ is an all-zero column $|\eta|$-vector. We readily see that the submatrix $[I_e : e \in \eta]$ of $I_\eta$ is the $|\eta| \times |\eta|$ identity matrix. Koetter and Médard [30] proved the following equality:

$$[f_e : e \in E] = K_{\rho} \cdot I_{\text{Out}^{G^\top}(\rho)} \cdot (I - K)^{-1}. \quad (77)$$

Then, for each sink node $\sigma_i \in S$ on $G^\top$, by (77), we have

$$F_{\sigma_i} = [f_e : e \in \text{In}^{G^\top}(\sigma_i)] = [f_e : e \in E] \cdot I_{\text{In}^{G^\top}(\sigma_i)}^{-1}. $$

10Here, we can see that the reverse order of “$\prec$” is a topological order on the reversed graph $G^\top$ that is consistent with the natural partial order of the edges in $E$ on $G^\top$. The use of this reverse order is convenient for the subsequent discussion.
Together with (76), we obtain that
\[
K_\rho \cdot I_{\text{Out}}^{\sigma_T}(\rho) \cdot (I - K)^{-1} \cdot I_{\text{In}}^{\sigma_T}(\sigma_i) \cdot K_{\sigma_i} = I_R, \\
\forall \sigma_i \in S. 
\]

(78)

Next, we construct an $F_q$-valued admissible $(R, 1)$ linear network code $C$ for $(\mathcal{N}, f)$ by “reversing” the decodable rate-$R$ linear network code $C_{\mathcal{N}^T}$ on $\mathcal{N}^T$. Now, we consider the original graph $G$.

- First, for each source node $\sigma_i \in S$, we let
  \[
  A_{i,e} \in \text{Out}(\sigma_i) = K_{\sigma_i}^T \quad (\text{cf. (40) for } A_{i,e}).
  \]
  Recall from (74) that $\text{Out}(\sigma_i) = \text{In} G^{\sigma_T}(\sigma_i)$, which immediately implies that
  \[
  A_{i,e} = K_{\sigma_i}^T \cdot I_{\text{In}}^{\sigma_T}(\sigma_i) 
  \]
  (cf. (43) for $A_{\sigma_i}$).
- For each intermediate node $v \in V \setminus (S \cup \{\rho\})$, we let $A_v = K_v^T$, i.e., let $a_{v,e} = k_{v,e}$ for all the adjacent pairs $(d, e)$ on $G$ with head$(d) = \text{tail}(e) = v$ (cf. (44) for $A_v$).

Now, by the above discussion, we have
\[
A = K^T
\]
(cf. (45) for $A$). For the sink node $\rho$ on $G$, we consider
\[
G(\sigma_i) \triangleq \left\{ g_{e, \sigma_i} : e \in \text{In}(\rho) \right\} = \left[ g_{e, \sigma_i} : e \in E \right] \cdot I_{\text{In}}^{\rho} = A_{\sigma_i} \cdot (I - A)^{-1} \cdot I_{\text{In}}^{\rho}
\]
\[
= K_{\sigma_i}^T \cdot I_{\text{In}}^{\sigma_T}(\sigma_i) \cdot (I - K)^{-1} \cdot I_{\text{Out}}^{\sigma_T}(\rho) \cdot K_{\rho} \quad \text{(81)}
\]
\[
= K_{\sigma_i}^T \cdot I_{\text{In}}^{\sigma_T}(\sigma_i) \cdot \left[ (I - K)^{-1} \cdot I_{\text{Out}}^{\sigma_T}(\rho) \cdot K_{\rho} \right] = I_R, \quad \text{(82)}
\]

since $\text{Out} G^{\sigma_T}(\rho) = \text{In}(\rho)$. Further, we obtain that
\[
G(\sigma_i) \cdot K_{\rho} = K_{\sigma_i}^T \cdot I_{\text{Out}}^{\sigma_T}(\rho) \cdot K_{\rho} = I_R \quad \text{(83)}
\]
where the last equality follows from the equality (78). Therefore, by considering all the source nodes $\sigma_i \in S$, we obtain that
\[
G(\sigma_i) \cdot K_{\rho} = \left[ G_{\sigma_i}^{(1)} \right] \cdot K_{\rho} = \left[ G_{\sigma_i}^{(2)} \right] \cdot K_{\rho} = \ldots = \left[ G_{\sigma_i}^{(m)} \right] \cdot K_{\rho} = [I_R]_m, \\
\]
which, together with (41), implies that the sink node $\rho$ computes the algebraic sum with zero error $R$ times by using the code $C$ once on the network $\mathcal{N}$, i.e., $C$ is an $F_q$-valued admissible $(R, 1)$ linear network code for the model $(\mathcal{N}, f)$.
Xuan Guang (Member, IEEE) received the Ph.D. degree in mathematics from Nankai University, Tianjin, China, in 2012. He was a joint Ph.D. student with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA, from January 2011 to August 2012. He was a Post-Doctoral Fellow under the Hong Kong Scholars Program with the Institute of Network Coding (INC), The Chinese University of Hong Kong, Hong Kong, SAR, China, from November 2015 to November 2018. Since September 2012, he has been with the School of Mathematical Sciences and the Key Laboratory of Pure Mathematics and Combinatorics (LPMC), Nankai University, where he is currently a Professor. His research interests include information theory, the areas of network coding, and network function computation. He was a recipient of the 2018 Chinese Information Theory Young Rising Star Award by China Information Theory Society.

Yang Bai (Student Member, IEEE) received the B.S. degree in mathematics from Nankai University, Tianjin, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Mathematical Sciences and LPMC. His research interests include information theory, network coding, network function computation, and information-theoretical security.

Raymond W. Yeung (Fellow, IEEE) was born in Hong Kong in June 1962. He received the B.S., M.Eng., and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, USA, in 1984, 1985, and 1988, respectively. He was on leave with Ecole Nationale Supérieure des Télécommunications, Paris, France, in Fall 1986. He was a member of Technical Staff of AT&T Bell Laboratories from 1988 to 1991. Since 1991, he has been with The Chinese University of Hong Kong, where he is currently a Choh-Ming Li Professor in information engineering and the Co-Director of the Institute of Network Coding. He has held visiting positions with Cornell University, Nankai University, the University of Bielefeld, the University of Copenhagen, Tokyo Institute of Technology, Munich University of Technology, and Columbia University. He was a consultant in a project of the Jet Propulsion Laboratory, Pasadena, CA, USA, for salvaging the malfunctioning Galileo spacecraft and a Consultant for NEC, USA. His 25-bit synchronization marker was used onboard the Galileo spacecraft for image synchronization. His research interests include information theory and network coding. He is the author of the textbooks A First Course in Information Theory (Kluwer Academic/Plenum 2002) and its revision Information Theory and Network Coding (Springer, 2008), which have been adopted by over 100 institutions around the world. This book has also been published in Chinese (Higher Education Press 2011, translation by Ning Cai et al.). He also coauthored with Shenghao Yang the monograph BATS Codes: Theory and Applications (Morgan & Claypool Publishers, 2017). In Spring 2014, he gave the first MOOC on information theory that reached over 25,000 students. He was a member of the Board of Governors of the IEEE Information Theory Society from 1999 to 2001. He was a recipient of the Croucher Foundation Senior Research Fellowship from 2000 to 2001, the 2005 IEEE Information Theory Society Paper Award, the Friedrich Wilhelm Bessel Research Award of the Alexander von Humboldt Foundation in 2007, the 2016 IEEE Eric E. Sumner Award (for pioneering contributions to the field of network coding), the 2018 ACM SIGMOBILE Test-of-Time Paper Award, the 2021 IEEE Richard W. Hamming Medal (for fundamental contributions to information theory and pioneering network coding and its applications), and the 2022 Claude E. Shannon Award. In 2015, he was named (together with Zhen Zhang) an Outstanding Overseas Chinese Information Theorist by the China Information Theory Society. In 2018, with Shenghao Yang he co-founded n-hop technologies in Hong Kong that has successfully deployed BATS code in the Hong Kong Government’s pilot smart lamppost system for wireless multi-hop transmission of sensor data. In 2019, his team won a Gold Medal with Congratulations of the Jury at the 47th International Exhibition of Inventions of Geneva for their invention “BATS: Enabling the Nervous System of Smart Cities.” He is a fellow of the Hong Kong Institution of Engineers, Hong Kong Academy of Engineering Sciences, and the U.S. National Academy of Inventors. He has served on the committees of a number of information theory symposiums and workshops. He was the General Chair of the First and the Fourth Workshops on Network, Coding, and Applications (NetCod 2005 and 2008), the Technical Co-Chair for the 2006 IEEE International Symposium on Information Theory, the Technical Co-Chair for the 2006 IEEE Information Theory Workshop (Chengdu, China), and the General Co-Chair of the 2015 IEEE International Symposium on Information Theory. He also serves as an Editor-at-Large for Communications in Information and Systems and an Editor for Foundation and Trends in Communications and Information Theory and Foundation and Trends in Networking. He was an Associate Editor for Shannon Theory of the IEEE TRANSACTIONS ON INFORMATION THEORY from 2003 to 2005. From 2011 to 2012, he serves as a Distinguished Lecturer of the IEEE Information Theory Society.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.