Simultaneously Generating Multiple Keys and Multi-Commodity Flow in Networks

Lifeng Lai
Department of Systems Engineering
University of Arkansas, Little Rock
Little Rock, AR 72204, USA
Email: lxlai@ualr.edu

Siu-Wai Ho
Institute for Telecommunications Research
University of South Australia
Adelaide, SA 5095, Australia
Email: siuwait.ho@unisa.edu.au

Abstract—The problem of simultaneously generating multiple independent keys for multiple pairs of users is considered. This problem is motivated by the fact that typically in wireless networks, multiple pairs of users need to establish secret keys for secure communications between these pairs. We propose a secure routing based key distribution approach to establish keys for the terminals. This approach connects the problem at the hand to that of multi-commodity flow problem studied in graph theory. Using the Max Bi-Flow Min Cut Theorem in the graph theory and developing a matching outer-bound, we show that the proposed approach achieves the key capacity region for the case of establishing two keys. For the general case, in which one needs to establish more than two keys, the achievable sum rate of the proposed approach achieves a constant factor to a derived upper-bound.

I. INTRODUCTION

Recently, a physical layer (PHY) approach to generate symmetric keys based on wireless channel reciprocity has attracted considerable attentions [1]–[4]. Here, the channel reciprocity refers to the case in which the channel response of the forward channel (from the transmitter to the receiver) is the same as the channel response of the backward channel (from the receiver to the transmitter) in time-division duplex (TDD) systems. Such a random channel serves as a common randomness source for the parties to generate keys. The eavesdroppers experience physical channels independent from the legitimate user’s channel as long as they are a few wavelength away from these legitimate nodes. Thus, the keys are provably secure with information-theoretic guarantee due to the nature of the key generation scheme, as opposed to the crypto keys whose security depends on the assumption of intractability of certain mathematical problems.

Until now, most of the existing work along this line considered the generation of a single key for a group of users. In particular, assuming that there are a set of wireless terminals $\mathcal{M}$, the problem of generating one common key for a group of terminals $A \subset \mathcal{M}$, with possible help from other wireless terminals $\mathcal{M}\setminus A$ using the public discussion, was studied in [5]. Furthermore, a pairwise independent network (PIN) model, which is particularly suitable for studying the key agreement problem in wireless networks, was proposed [6], [7]. This model, as a special case of the general model considered in [5], was motivated by the fact that each pair of wireless terminals can obtain correlated estimates of the channel gain between them. This pair of estimates are independent of estimates associated with the channel gains from other channels. Hence, the name of the PIN was used. The generation of a single key from this PIN model was studied in [6] using tools from graph theory. In particular, [6] proposed an effective key propagation approach that connects the key generation problem to the Steiner tree packing problem in a multigraph. This approach is shown to be optimal for the special case of $|A| = 2$ or $A = \mathcal{M}$ [6].

In this paper, we consider the simultaneous generation of multiple keys, each for a pair of users, under the PIN model. This is motivated by the fact that there are typically multiple pairs of nodes communicating with each other in wireless networks. Each pair of nodes need to establish a key between them so that they can use their respective secret key for encryption and decryption. Under the model studied, there are a set of terminals $\mathcal{M}$, among which $T$ pairs of terminals want to generate $T$ independent keys with the assistance of the remaining users. Clearly, there are tradeoffs among the rates of generating these $T$ keys. We are interested in characterizing the key rate region. We propose a simple approach to propagate the keys through the network. In the proposed approach, we first construct a graph for the PIN model. In the graph constructed, the set of nodes is the same as the set of wireless terminals, and the link capacity between nodes $i$ and $j$ is the same as the rate of the mutual information between the local channel estimates. The terminals then establish routes between the terminals that need to establish common keys. Using these routes, one of each pair of terminals that are involved in establishing a common key then sends randomly generated keys to the other terminal involved. Along each route, this randomly generated key will be encrypted and decrypted using local key established via local correlated estimates. This secure routing approach effectively converts the simultaneous key establishment problem to a multi-commodity flow problem in a multigraph [8]. By deriving an outer bound on the rate region coupled with results from the graph theory, we show

The work of L. Lai was supported by the National Science Foundation CAREER Award under Grant CCF-1054338 and by the National Science Foundation under Grant CNS-1116534. The work of S.-W. Ho was supported by the Australian Research Council under an Australian Postdoctoral Fellowship as part of Discovery Project DP1094571.
that the proposed key propagation approach is optimal for simultaneously generating two keys for two pairs of nodes. We then fully characterize the rate region for this case. We also extend the study to the case of simultaneously generating more than 2 keys. In this general case, we show that the proposed approach achieves a sum-rate that is constant factor to that of an upper-bound derived in the paper.

The remainder of this paper is organized as follows. In Section II, we give details about the model discussed in this paper. In Section III, we characterize the key rate region for the case of generating two keys. In Section IV, we consider the general case of generating more than two keys. In Section V, we offer some concluding remarks.

II. MODEL

We follow the notations established in [6]. There are \( m \) terminals, indexed using \( M = \{1, \cdots, m\} \). Each terminal \( i \in M \) observes \( X^n_i = (X_{1i}, \cdots, X_{ni}) \), which are \( n \) independent and identically distributed (i.i.d.) repetitions of the \( X_i \). Same as [6], we consider the PIN model. More specifically, each \( X_i \) is of the form \( X_i = (X_{ij}, j \in M \setminus \{i\}) \), and the pairs \( \{ (X_{ij}, X_{ji}), 1 \leq i < j \leq m \} \) are mutually independent. The goal is to generate \( T \) independent keys \( \{K_t, t = 1, \cdots, T\} \), one for each pair of users indexed by \( (t, T+t), t = 1, \cdots, T \). All the remaining users ranging \( 2T+1, \cdots, m \) serve as helpers that are not required to recover any of the keys nor they are required to be kept secret from these keys. To fulfill this goal, the terminals in \( M \) are allowed to exchange information over a public channel with unlimited capacity. However, all the information exchanged over the public channel will be overheard by the Eve. We use \( F \) to denote the information exchanged over the public channel. Combining \( X^n_i \) and \( F \), terminal \( t \) generates an estimate \( K_t \) of the key \( K_i \) for \( 1 \leq t \leq T \) (or \( K_{t-T} \) for \( T+1 \leq t \leq 2T+1 \)), where \( K_i \) is defined on an alphabet \( K_t \). We have the following requirements regarding the keys:

\[
\begin{align*}
\Pr\{K_t = \hat{K}_t = K_{t+T}\} & \geq 1 - \epsilon, \forall t \in \{1, \cdots, T\}, \quad (1) \\
I(K_t; K_j) & = 0, \forall i \neq j, \forall i, j \in \{1, \cdots, T\}, \quad (2) \\
\log |K_t| - H(K_t) & \leq \epsilon, \forall t \in \{1, \cdots, T\}. \quad (3) \\
I(K_t; F) & \leq \epsilon, \forall t \in \{1, \cdots, T\}. \quad (4)
\end{align*}
\]

Here, (1) means that the pair \( (t, t+T) \) generates the same key, (2) implies that these \( T \) keys are mutual independent, (3) says that the keys are close to be uniformly generated, and (4) implies that Eve learns limited amount of information about the generated keys.

A rate vector \((R_1, \cdots, R_T)\) is said to be achievable, if there exists communication strategy \( F \) such that conditions (1)-(4) are satisfied and

\[ R_t = \frac{1}{n} \log |K_t|. \quad (5) \]

The set of all achievable rate vectors is called the capacity region. Furthermore, we are also interested in the maximal sum of key rates

\[ C_{\text{sum}} = \sup_{t=1}^T R_t. \quad (6) \]

III. TWO PAIRS CASE

We first consider the case of \( T = 2 \), i.e., we need to generate key \( K_1 \) for terminals \((1,3)\) and key \( K_2 \) for terminals \((2,4)\). All other terminals serve as helpers that will assist in the key generation process. They are not required to recover the value of keys, nor they are required to be kept secret from the generated keys. Let \( B \) be a subset of \( M \) and \( B' = M \setminus B \). We say that a user pair \((i,j)\) crosses \( \{B,B'\} \) if either \( i \in B \) and \( j \in B' \) or \( i \in B' \) and \( j \in B \). We first provide an outer bound on the key rate region.

**Theorem 1:** A rate pair \((R_1, R_2)\) is achievable only if the following conditions are satisfied:

\[
\begin{align*}
R_1 & \leq \min_{B_1 \subseteq B_1, 3 \notin B_1} \sum_{(i,j) \in B_1} I(X_{ij}; X_{ji}), \quad (7) \\
R_2 & \leq \min_{B_2 \subseteq B_2, 4 \notin B_2} \sum_{(i,j) \in B_2} I(X_{ij}; X_{ji}), \quad (8) \\
R_1 + R_2 & \leq \min_{B_3, B_4} \sum_{(i,j) \in B_3 \cup B_4} I(X_{ij}; X_{ji}). \quad (9)
\end{align*}
\]

**Proof:** The basic idea of the proof is to construct a genieaid model and show that the capacity region of this genie-aid model is upper bounded by (7)-(9). The details are omitted due to space limitation.

In the following, we describe a graph-based approach that allows nodes to propagate keys over the network. There are two main steps: 1) graph construction via local key establishment; and 2) key propagation via multi-commodity flow.

**Algorithm 1: Key Propagation Algorithm**

- **Step 1: Graph Construction:** Construct a graph \( G_n(V,E) \), in which \( V \) and \( E \) are the set of nodes and edges of the graph respectively. In our graph, \( V \) includes all the nodes in \( M \). For each node pair \((i,j)\), we add an undirected secure link with link capacity \( e_{ij} = n(I(X_{ij}; X_{ji}) - \epsilon) \). This is done by asking node \( i \) and \( j \) to establish a local key via the existing point-to-point key establishment protocol with the correlated observations \( (X^n_{ij}, X^n_{ji}) \) [9]. We use \( K_{ij} \) to denote the value of this local key at node \( i \) and \( K_{ji} \) to denote the value of this key at node \( j \). We have that

\[ \Pr\{K_{ij} \neq K_{ji}\} \leq \epsilon. \quad (10) \]

In the following, instead of using both \( K_{ij} \) and \( K_{ji} \) to denote the value of the local key between \((i,j)\), we will use \( K_{ij} \) to denote both keys with the understanding that there is a small probability that the value of local keys at \((i,j)\) are different. We use \( F_{ij} \) to denote the public
discussion information exchanged in order to establish the local key between \((i, j)\). From [5], [9], we know that there exists a scheme such that
\[
I(K_{ij}; F_{ij}) \leq \epsilon_2. \tag{11}
\]

- **Step 2: Key Propagation:**
  Nodes 1 and 2 randomly generate keys \(K_1\) and \(K_2\) from sets \(\{1, \ldots, 2^nR_1\}\) and \(\{1, \ldots, 2^nR_2\}\) using a uniform distribution. Hence, \(K_1\) has \(nR_1\) bits while \(K_2\) has \(nR_2\) bits. Node 1 then sends these \(nR_1\) bits of information to node 3 using a secure routing approach. More specifically, let \(P_l^1 = (1, i_l, i_{l-1}, \ldots, 3)\) be the \(l\)th route between node 1 and node 3, and \(Q_l^1\) be the total number of hops in this route. Node 1 divides key \(K_1\) into \(L_1\) non-overlapping parts \(\{K_{11}, K_{12}, \ldots, K_{1L_1}\}\), each having length \(W_1\) bits, and sends \(K_{1l}\) through the \(l\)th route. In the \(q\)th hop of the \(l\)th route \((i_l, q, i_{l-1})\), node \(i_l,q\) encrypts \(K_{1l}\) using \(W_l\) bits of the local key \(K_{i_l,q,i_{l-1}}\). We use \(K_{i_l,q,i_{l-1}}\) to denote this part of the local key. In this case, node \(i_l,q\) uses the one-time pad scheme for encryption, namely node \(i_l,q\) broadcasts \(K_{1l} \oplus K_{i_l,q,i_{l-1}}\) over the public channel. After that, node \(i_l,q+1\) decrypts \(K_{1l}\) using the same part of the local key \(K_{i_l,q,i_{l-1}}\), namely \(K_{1l,i_l,q,i_{l-1}}\). After that, the node pair \((i_l,q, i_{l,q+1})\) will discard \(K_{1l,i_l,q,i_{l-1}},\) i.e., \(K_{1l,i_l,q+1,i_{l-2}}\), which will not be used again. Similarly, node 2 divides key \(K_2\) into \(L_2\) parts, and send them to the node 4 using one-time pad through \(L_2\) different secure routes, each having \(Q_2^l\) hops.

In Fig. 1, we show an example of a network consisting of 5 nodes. Several routes are shown in the figure. In this case, node 1 randomly generates a key \(K_1\) and divides it into three parts \(\{K_{11}, K_{12}, K_{13}\}\), which will be sent over routes \((1, 3)\), \((1, 5, 3)\), \((1, 2, 5, 3)\) to node 3 respectively. Node 2 randomly generates a key \(K_2\) and divided into two parts \(\{K_{21}, K_{22}\}\), which will be sent over routes \((2, 4)\), \((2, 5, 4)\) to node 4 respectively. When \(K_{21}\) is sent over the route \((1, 3)\), node 1 will send \(K_{11} \oplus K_{13}\). The pair node \((2, 5)\) will divide its local key \(K_{25}\) into two parts, one is used to encrypt the third part of \(K_1\), namely \(K_{13}\), the other one is used to encrypt the second part of key \(K_2\), namely \(K_{22}\). To ensure the secrecy of the randomly generated keys \(K_1\) and \(K_2\), we require that the total amount of key parts over each edge is less than the amount of the locally established point-to-point key.

**Theorem 2:** The key generation approach specified in Algorithm 1 satisfies conditions (1)–(4).

**Proof:** We first look at the probability that the value of key \(K_1\) recovered at node 3 is different from \(K_1\). The only type of event that may lead to a decoding error at node 3 is an error during the local key establishment process. More specifically, suppose the link \((i, j)\) is one of the routes used by node 1, then the event \(K_{ij} \neq K_{ij}\) may lead to an error of the key recovered at node 3. We have
\[
Pr[K_1 \neq K_1] \leq Pr[\exists(i, j)\text{ such that } K_{ij} \neq K_{ij}] 
\leq \binom{m}{2} \epsilon_1. \tag{12}
\]
Here, we have used the union bound. This probability can be made arbitrarily small. Similarly \(Pr[K_2 \neq K_2]\) can be made arbitrarily small.

We also note that (2) and (3) are satisfied since both \(K_1\) and \(K_2\) are independently generated using the uniform distribution.

In the following, we compute the amount of key leakage. Two types of information has been exchanged over the public channel: 1) \(\{F_{ij}, 1 \leq i < j \leq m\}\) that are used to establish local keys; 2) \(\{K_{1l} \oplus K_{i_l,q;i_{l-1}}, 1 \leq l \leq L_1, 1 \leq q \leq Q_1^l\}\) that are used to route \(K_{i_l,q;i_{l-1}}\), \(1 \leq l \leq L_1\) from node 1 to node 3. We have
\[
I(K_1; \{F_{ij}, 1 \leq i < j \leq m\}, \{K_{1l} \oplus K_{i_l,q;i_{l-1}}, 1 \leq l \leq L_1, 1 \leq q \leq Q_1^l\})  
= I(K_1; \{F_{ij}, 1 \leq i < j \leq m\}) + I(K_1; \{K_{1l} \oplus K_{i_l,q;i_{l-1}}, 1 \leq l \leq L_1, 1 \leq q \leq Q_1^l\} | \{F_{ij}, 1 \leq i < j \leq m\}) \tag{13}
\]
\[
= H(\{K_{1l} \oplus K_{i_l,q;i_{l-1}}, 1 \leq l \leq L_1, 1 \leq q \leq Q_1^l\})  
- H(\{F_{ij}, 1 \leq i < j \leq m\}) \tag{14}\]
\[
\leq H(\{K_{i_l,q;i_{l-1}}, 1 \leq l \leq L_1, 1 \leq q \leq Q_1^l\}) + \max\{Q_1^l\} L_1 \epsilon \tag{a}
\]
\[
\leq \max\{Q_1^l\} L_1 (\epsilon + \epsilon_2), \tag{15}
\]
where (a) is due to (3). The bound in (15) can be made arbitrarily small as \(n\) increases. This implies that Eve learns negligible amount of information about the generated key \(K_1\) from the public discussion and subsequent key routing process. Similarly, one can show that the public discussion and key routing reveals negligible amount of information about \(K_2\).

It is clear that the proposed secure routing key propagation protocol converts the simultaneous key agreement problem into a multi-commodity flow problem over the graph \(G_n(V,E)\) [8]. In this equivalent multi-commodity flow problem, we have two commodities that need to be transferred from node 1 to node 3 and from node 2 to node 4 with the constraint that the total amount of flows on each link cannot exceed the flow capacity. Maximizing the achievable key rates using this
approach is the same as maximizing the rates of these two flows by carefully selecting the routes and the amount of flow over each route. In the following, we show that by suitable routes, this secure routing approach achieves the outer bound specified in Theorem 1.

Theorem 3: The scheme in Algorithm 1 achieves the upper-bound specified in Theorem 1 and hence is optimal.

Proof: The proof relies on the Max Bi-flows Min-Cut Theorem in the graph theory established in [10]. We use \( f(1,3) \) to denote the amount of flow between node 1 and 3, and use \( f(2,4) \) to denote the amount of flow between node 2 and 4 on graph \( G_n(V,E) \). From [10], we know that as long as

\[
f(1,3) \leq \min_{1 \in B_1, 3 \in B_3} \sum_{(i,j) : i \in B_1, j \in B_3} e_{ij},
\]

\[
f(2,4) \leq \min_{2 \in B_2, 4 \in B_4} \sum_{(i,j) : i \in B_2, j \in B_4} e_{ij},
\]

\[
f(1,3) + f(2,4) \leq \min_{(1,3) \in B_1, (2,4) \in B_2} \sum_{(i,j) : i \in B_3, j \in B_4} e_{ij},
\]

\[
f(1,3) + f(2,4) \leq \min_{(1,4) \in B_1, (2,3) \in B_2} \sum_{(i,j) : i \in B_3, j \in B_4} e_{ij},
\]

one can construct routes and corresponding flows on each route that allow \( f(1,3) \) amount of flow from node 1 to node 3 and \( f(2,4) \) amount of flow from node 2 to node 4. Plugging

\[
f(1,3) = nR_1,
\]

\[
f(2,4) = nR_2,
\]

\[
e_{ij} = n(I(X_{ij}, X_{ji}) - \epsilon),
\]

into (16)–(19), we know that the secure routing based key propagation approach achieves the outer bound established in Theorem 1.

One can use the cycle flow method proposed in [10] to efficiently find the routes and the corresponding flows that achieve the capacity region. The basic idea is to recursively construct routes for each user under the rate constraints.

IV. GENERAL CASE

We now consider the general case in which we are required to generate \( T > 2 \) keys, one key for each pair \((t, t+T)\), \( t = 1, \ldots, T \). In this general case, we discuss the sum of key rates. We will generalize Algorithm 1 to this general case. We will also provide an upper-bound on the sum rate, and show that the sum rate achieved using the routing based key propagation approach is a constant fact away from the developed upper-bound.

The secure routing-based key propagation scheme discussed in Section III can be used in this general scenario. In particular, we again construct a graph \( G_n(V,E) \) with \( V \) being the same as \( M \) and \( E \) being the set of edge of capacity \( e_{ij} = n(I(X_{ij}, X_{ji}) - \epsilon) \). Node \( t, 1 \leq t \leq T \) then randomly generates a key \( K_t \) using a uniform distribution from the set \( \{1, \ldots, 2^{nR_t}\} \). Node \( t \) further divides \( K_t \) into non-overlapping \( L_t \) parts \( \{K_{t,1}^1, \ldots, K_{t,L_t}^1\} \), where each part is sent over a route from node \( t \) to node \( t+T \). During the routing, each key part \( K_{t,i}^1 \) is encrypted and decrypted using the local keys established from the pair-wise correlated observations. Following the same steps in the proof of Theorem 2, one can show that as long as the sum of key parts flow through each edge \((i, j)\) is less than the edge capacity \( e_{ij} \), there is arbitrarily small error probability of key recovery and Eve can learn negligible amount of information about the established keys. It is clear that this routing based approach converts the problem into a multi-commodity flow problem in the graph \( G_n(V,E) \). Finding the maximum achievable sum of rates \( C_r \) using this approach is equivalent to finding the maximum sum of the rates of multi-commodity flows, which has been extensively studied in the graph theory. In particular, one can formulate a linear programming to characterize \( C_r \). Furthermore, many efficient approximation algorithms have been developed [8].

We now develop an upper-bound for the sum rate of key rates for any key generate protocol. We will use a graph \( G_n^*(V,E) \) that is the same as \( G_n(V,E) \) constructed above with a modification that \( e_{ij} = nI(X_{ij}; X_{ji}) \). A set of edges \( E' \) of the graph \( G_n^*(V,E) \) is called a multicut if removing the set \( E' \) from the graph \( G_n^*(V,E) \) disconnects node \( t \) from \( t+T \) for \( t = 1, \ldots, T \). Equivalently, a set \( E' \) is a multicut if for all \( t = 1, \ldots, T \), there is no path between node \( t \) and \( t+T \) in the graph \( G_n^*(V,E \setminus E') \). This implies that a multicut \( E' \) divides the set of nodes \( V \) into \( U \) non-overlapping subsets \( V_1, V_2, \ldots, V_U \) such that for all \( t = 1, \ldots, T \), node \( t \) and node \( t+T \) are in two different subsets. It is easy to see that \( U \leq m \). For each node set \( V_u \) with \( u = 1, \ldots, U \), we define a set \( E_{V_u}' \subset E' \) such that an edge \((i, j) \in E' \) is in the set \( E_{V_u}' \) if either \( i \in V_u \) or \( j \in V_u \). Clearly each edge \((i, j) \in E' \) belongs to two different \( E_{V_u}' \)'s. Figure 2 illustrates a multicut and the associated definitions. The value of a multicut \( E' \) is defined as

\[
C_{E'} = \sum_{(i,j) \in E'} e_{ij} = \sum_{(i,j) \in E'} nI(X_{ij}; X_{ji}).
\]
Theorem 4:

\[ C_{\text{sum}} \leq \frac{1}{n} \min_{E'} C_{E'} = \min_{E'} \sum_{(i,j) \in E'} I(X_{ij}; X_{ji}). \tag{24} \]

Proof: As discussed above for any given multicut \( E' \), there is an associated node partition \( V_1, \ldots, V_U \). For each such \( V_u \), let \( V_u^{\text{key}} = V_u \cap \{1, 2, \ldots, 2T\} \), that is \( V_u^{\text{key}} \) is the set of nodes that are required to generate keys and are in \( V_u \). We also use \( V_u^{\text{key,c}} \) to denote the set of nodes that consist key generation pairs with nodes in \( V_u^{\text{key}} \). Since \( E' \) is a multicut, \( V_u^{\text{key,c}} \subset V_u^{\text{c}} \). Here \( V_u^{\text{c}} \) is the complimentary set of \( V_u \), i.e., \( V_u^{\text{c}} = V \setminus V_u \). It is easy to see that

\[ E_{V_u}^{'} = \{(i, j) : i \in V_u, j \in V_u^{\text{c}}\}. \tag{25} \]

For any \( V_u \) of a given multicut \( E' \), we consider a genie-aided model created as following. We first create a supernode \( V_u^{*} \) by combining all observations at nodes in \( V_u^{\text{key}} \) and another supernode \( V_u^{*,c} \) by combining all observations at nodes in \( V_u^{\text{key},c} \). Now, in this modified model, our goal is to generate only a shared key for the two super nodes \( V_u^{*} \) and \( V_u^{*,c} \), and all other nodes act as helpers for this purpose. We use \( R_{V_u} \) to denote the largest rate possible for this modified model. Following similar arguments in Theorem 1, we know that

\[ nR_{V_u} \leq \sum_{(i,j) \in E_{V_u}^{'}} nI(X_{ij}; X_{ji}). \tag{26} \]

Clearly, any key generation protocol for the original model with key rate \( R_i \) for \( i \in V_u^{\text{key}} \) can be used to generate a key for this two super nodes in this genie aided model\(^1\). Hence

\[ n \sum_{i \in V_u^{\text{key}}} R_i \leq nR_{V_u} \leq \sum_{(i,j) \in E_{V_u}^{'}} nI(X_{ij}; X_{ji}). \tag{27} \]

One can repeat the same steps as above for each \( V_u, 1 \leq u \leq U \), and (27) is true for each \( u \).

Now summing over these \( U \) partitions associated with the multicut \( E' \), we have

\[ n \sum_{u=1}^{U} \sum_{i \in V_u^{\text{key}}} R_i \leq \sum_{u=1}^{U} \sum_{(i,j) \in E_{V_u}^{'}} nI(X_{ij}; X_{ji}). \tag{28} \]

Noting that

\[ \sum_{i=1}^{T} R_i = 2 \sum_{i=1}^{T} R_i, \tag{29} \]

\[ \sum_{i=1}^{T} \sum_{(i,j) \in E_{V_u}^{'}} nI(X_{ij}; X_{ji}) = 2C_{E'}, \tag{30} \]

we have

\[ n \sum_{i=1}^{T} R_i \leq C_{E'}. \tag{31} \]

\(^1\)Here for notation convenience, we allow \( i \) to be in the range of 1 and 2\( T \) with the understanding that \( R_i = R_{i-T} \) if \( i > T \).

Since (31) is true for any multicut \( E' \), we have

\[ C_{\text{sum}} \leq \frac{1}{n} \min_{E'} C_{E'} = \min_{E'} \sum_{(i,j) \in E'} I(X_{ij}; X_{ji}). \tag{32} \]

The following theorem characterize the relationship between the sum rate achieved using our routing based approach to that of the upper-bound derived in Theorem 4.

Theorem 5:

\[ C_r \geq C_{\text{sum}}/O(\log T). \tag{33} \]

Proof: The proof is an application of a result in graph theory that characterizes the relationship between max sum flow and min multi-cut [11] of a graph. More specifically, for a graph \( G_n(V, E) \), using Theorem 5.1 of [11], we have

\[ nC_r \geq \frac{1}{O(\log T)} \min_{E'} C_{E'}. \tag{34} \]

Coupled with (24), we have the desired result.

\[ \blacksquare \]

V. CONCLUSION

The problem of simultaneously generating multiple keys has been considered under the PIN model. A simple secure routing based key propagation protocol has been proposed. This approach converts the problem under study to a multi-commodity flow problem in networks. We have shown that the proposed approach is optimal for the case of generating two keys. We have also shown that the sum rate of the proposed scheme is a constant fraction away from a derived upper-bound.