Key Generation Algorithms for Pairwise Independent Networks Based on Graphical Models

Lifeng Lai, *Member, IEEE*, and Siu-Wai Ho, *Senior Member, IEEE*

***Abstract*— We consider two secret key generation problems under a pairwise independent network model, and propose low complexity key generation schemes in a framework that connects our problems to network flow problems in graphs. Our schemes have two components: 1) local key generation and 2) global key propagation. In the local key generation, we use point-to-point source coding with side information to establish pairwise keys, from which we construct a graph with the capacity of each edge being the key rate of the corresponding point-to-point local key. In the global key propagation, depending on the particular problem, secret keys are delivered to users in the network using various network flow algorithms. In particular, in the first problem in which one is required to generate a group key for a group of users in the network, we propose a network coding-based global key propagation approach. This approach has a low complexity and has a better performance than the existing approach. In the second problem, in which one is required to generate multiple keys simultaneously for different pairs of users, we propose a multicommodity flow-based global key propagation approach. We show that the proposed approach is optimal for the case of generating two keys. For the general case of generating more than two keys, we show that the sum rate of the proposed scheme is larger than an upper bound characterized in this paper divided by a constant.**

***Index Terms*— Graphical models, key generation, multi-commodity flow, network coding, routing.**

I. Introduction

ESTABLISHING a symmetric key to be shared by a pair of users in secure communications is a challenging task. Interestingly, in a so called key generation via public discussion approach, [2], [3] showed that two terminals, namely Alice and Bob, can generate secret keys by talking to each other in public, as long as these two terminals share correlated but not necessarily same random variables. The basic idea of this approach is to use the technique of source coding with side information. Roughly speaking, Alice can divide all possible sequences it observes into bins and reveal the bin number to Bob via the public discussion. By combining its own observations and the bin number sent by Alice, Bob will be able to recover the sequence observed by Alice with a high probability. It can be shown that the bin index tells little information about the index of the sequence within the bin. Hence, both Alice and Bob can use the index within the bin as the generated key. As the result, one can adopt existing practical Slepian-Wolf codes [4] to construct practical schemes for the key generation [5] in this point-to-point symmetric key generation setup.

Reference [6] further extended the study to a group key generation problem in a general network setup. Among other scenarios, Csiszár and Narayan [6] discussed a scenario in which there is a set of users $\mathcal{M}$, among which a subset of users in $A \subseteq \mathcal{M}$ would like to establish a single group key to be shared by the users in A with possible assistance from the remaining users $\mathcal{M} \setminus A$. Reference [6] identified a close connection between the group key generation problem and distributed source coding problems, and fully characterized the secret key capacity for general source models. In particular, Csiszár and Narayan [6] showed that the secret key capacity is the joint entropy of source observations of all users in $\mathcal{M}$ minuses the minimum sum distributed source coding rates that enable users in A to recover the random variables of all users in $\mathcal{M}$. The structure of the solution was further studied in [7].

However, unlike the point-to-point scenario, very few practical schemes for network distributed source coding exist. By focusing on a special source model named pairwise independent network (PIN) model,[1] Nitinawarat *et al.* [16] proposed an interesting scheme that converts the group key generation problem [6] into a combination of 1) local pair-wise key generation; and 2) global key propagation.

Manuscript received April 8, 2013; revised August 24, 2014; accepted April 5, 2015. Date of publication June 29, 2015; date of current version August 14, 2015. L. Lai was supported in part by the Division of Computing and Communication Foundations through the National Science Foundation (NSF) CAREER Award under Grant CCF-1318980 and in part by the Division of Computer and Network Systems through NSF under Grant CNS-1321223. S.-W. Ho was supported by the Australian Research Council through the Australian Post-Doctoral Fellowship under Project DP1094571. This paper was presented at the 2012 IEEE Information Theory Workshop [1].

L. Lai is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: llai@wpi.edu).

S.-W. Ho is with the Institute for Telecommunications Research, University of South Australia, Adelaide, SA 5095, Australia (e-mail: siuwai.ho@unisa.edu.au).

Communicated by U. M. Maurer, Associate Editor for Complexity and Cryptography.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2015.2450729

[1] The PIN model first studied in [8] is well motivated by real life scenarios. In particular, it is particularly suitable for studying the key agreement problem in wireless networks [9]–[15]. This model, as a special case of the general model considered in [6], was motivated by the observation 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.

With this approach, one can then again take advantage of the existing practical Slepian-Wolf codes in the first step. The scheme in [16] achieves the key capacity when $|A| = 2$ (i.e., only two nodes are required to generated the key) or when $A = \mathcal{M}$ (i.e., all nodes are required to recover the group key.) However, in the general case, there are some challenges in the second step of the approach in [16]. In particular, finding the best global key propagation pattern in the second step is equivalent to the Steiner tree packing problem in a multigraph [17], which is NP-complete.

In this paper, we extend the work in [16] along two lines with the goal of designing schemes that achieve better key rates while avoiding the complexity issues associated with the Steiner tree packing problem or other combinational optimization problems. The proposed schemes are combinations of point-to-point key generation problem, for which practical coding schemes exist, and low complexity linear programming (LP) problems. They outperform the existing schemes and are optimal in certain scenarios. Our approach draws connections between key generation problems and various problems in graph theory, which allow us to utilize rich tools and literature in graph theory.

In the first line, we consider the generation of a *single group* key under the PIN model, the same setup as [16]. We propose a low complexity scheme that outperforms the approach in [16]. The enabling element of our scheme is a close connection between the group key generation and the multicast over graph problem. 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 $\mathcal{M}$, and the link capacity between nodes i and j is the same as the rate of the mutual information between the source observations at these two nodes. We then construct a network code that achieves the largest multicast throughput from node 1 to the set of users in A for the graph constructed. Node 1 then randomly generates a key and multicasts this key to other users in A using network coding. At each hop, the information will be encrypted and decrypted using local key established during the graph construction phase. This network coding based approach allows us to tap into a vast amount of existing work on network coding for multicast, and allows us to achieve a better key rate than the equivalently pure routing-based approach in [16]. Furthermore, finding the largest achievable rate under the proposed approach is essentially a linear programming (LP) problem, which can be solved efficiently. We also note that network coding has been used for the key generation problem in [18] for a different model. In particular, in [18], the sources observed at the nodes are linear combinations of a common set of independent random variables that are uniformly distributed over a common finite field. This special source structure enables one to use network coding for the key generation problem considered in [18]. We note that the PIN model considered in our paper does not satisfy the structure required in [18]. Hence, the approach in [18] is not applicable in our case.

In the second line, we extend the study to the simultaneous generation of *multiple* keys, each for a different 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 communication networks. Each pair of nodes needs 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, same as the first scenario. 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 graph [17]. By deriving an outer bound on the rate region coupled with results from graph theory, we show 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 two 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.

In summary, compared with the two existing interesting work [16] and [18] that are most relevant to our paper, our contributions are:

1) The single key generation problem: This problem has been considered in [16] and [18]. Compared with the linear model in [18], we consider the PIN model, which is well-motivated by key generation over wireless channels and is less restrictive. Compared with the scheme in [16], we show that in the PIN model, network coding can achieve better key rates. Furthermore, while finding the largest key rate using the approach in [16] is a NP-complete problem, finding the largest achievable rate under the proposed approach is essentially a linear programming (LP) problem, which can be solved efficiently.
2) The multiple key generation problem: In Section IV, we study a novel model of generating multiple keys simultaneously. We propose a simple secure routing approach for generating multiple keys. This secure routing approach effectively connects the simultaneous key establishment problem to a multi-commodity flow problem in a graph. By deriving an outer bound on the rate region coupled with results from graph theory, we show that the proposed key propagation approach is optimal for simultaneously generating two keys for two pairs of nodes. We fully characterize the rate region for this case. We also extend the study to the case of simultaneously generating more than two keys. In this general case, we show that the maximum sum-rate can be characterized by a LP. Furthermore, we show that the

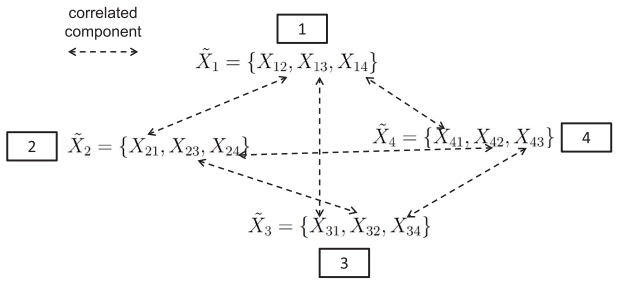

Fig. 1. PIN model for 4 nodes.

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. Section III presents our results for the single group key generation problem. In Section IV, we present our results for the generation of multiple keys. In Section V, we offer some concluding remarks.

II. Model

We follow the notations established in [16]. There are m terminals, indexed using $\mathcal{M} = \{1, \cdots, m\}$. Each terminal $i \in \mathcal{M}$ observes $\tilde{X}_i^n = (\tilde{X}_{i1}, \cdots, \tilde{X}_{in})$, which are n independent and identically distributed (i.i.d.) repetitions of $\tilde{X}_i$. Same as [16], we consider the PIN model. More specifically, each $\tilde{X}_i$ is of the form $\tilde{X}_i = \{X_{ij}, j \in \mathcal{M} \setminus \{i\}\}$, and the pairs $\{(X_{ij}, X_{ji}), 1 \leq i < j \leq m\}$ are mutually independent. As the result, in the PIN model,

$$P_{\tilde{X}_1, \cdots, \tilde{X}_m}(\tilde{x}_1, \cdots, \tilde{x}_m) = \prod_{1 \leq i < j \leq m} P_{X_{ij}, X_{ji}}(x_{ij}, x_{ji}). \quad (1)$$

Figure 1 illustrates the PIN model when $m = 4$.

These users are allowed to exchange information with each other using a public channel with infinite capacity. However, any information exchanged using the public channel will also be perfectly received by Eve. Without loss of generality, one can assume that these users take turn in sending public information for r rounds. We use $f_1, \cdots, f_{rm}$ to denote the public information exchanged. Here, f_s is the information sent by user $i = (s \mod m) + 1$ at time s. f_s depends on the random sequences $\tilde{X}_i^n$ at user i and the public discussion $f_1, \cdots, f_{s-1}$ that has occurred so far. We use $\mathbf{F}_j$ to denote the collection of public discussion sent by j during the public discussion phase, and $\mathbf{F} = [\mathbf{F}_1, \cdots, \mathbf{F}_m]$ to denote the whole set of public discussion. Eve knows the functions used by each user for generating the public information, and knows $\mathbf{F}$ perfectly.

We consider two scenarios: 1) to generate a single group key for a group of users; and 2) to generate multiple independent keys, each for a pair of users.

A. Single Group Key Generation

In the first scenario, we consider the generation of a single group key for a group of users in $A \subseteq \mathcal{M}$, the same setup as [16]. Without loss of generality, we assume that $A = \{1, \cdots, a\}$, and all the remaining users indexed by $\{a+1, \cdots, m\}$ act as helpers that are not required to recover any of the keys nor they are required to be kept secret from these keys. At the end of the public discussion, by combining $\tilde{X}_i^n$ and $\mathbf{F}$, each user $i \in A$ generates a key $\hat{K}_i$. For any $\epsilon > 0$, we require that

$$\Pr\{K = \hat{K}_1 = \hat{K}_2 = \cdots = \hat{K}_a\} \geq 1 - \epsilon, \quad (2)$$

$$\log |\mathcal{K}| - H(K) \leq \epsilon, \quad (3)$$

$$I(K; \mathbf{F}) \leq \epsilon. \quad (4)$$

Here, (2) implies that the users in group A generate the same key K with a high probability, (3) implies that the generated key is nearly uniformly distributed and (4) implies that Eve learns a limited amount of information about the generated key from the public discussion.

From the set $\{\tilde{X}_i^n\}$, a rate R is said to be achievable, if there exists a public discussion strategy $\mathbf{F}$ such that conditions (2)–(4) are satisfied and

$$R = \frac{1}{n} H(K), \quad (5)$$

as $n \to \infty$. We call the largest achievable key rate as the key capacity C.

B. Multiple-Key Generation

In this scenario, our 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. Combining $\tilde{X}_t^n$ and $\mathbf{F}$, terminal t generates an estimate $\hat{K}_t$ of the key K_t for $1 \leq t \leq T$ (or K_{t-T} for $T + 1 \leq t \leq 2T$), where K_t is defined on an alphabet $\mathcal{K}_t$. For any $\epsilon > 0$, we have the following requirements regarding the keys:

$$\Pr\{K_t = \hat{K}_t = \hat{K}_{t+T}\} \geq 1 - \epsilon, \ \forall t \in \{1, \cdots, T\}, \quad (6)$$

$$H(K_1, K_2, \ldots, K_T) = \sum_{t=1}^{T} H(K_t), \quad (7)$$

$$\log |\mathcal{K}_t| - H(K_t) \leq \epsilon, \ \forall t \in \{1, \cdots, T\}. \quad (8)$$

$$I(K_1, \cdots, K_T; \mathbf{F}) \leq \epsilon. \quad (9)$$

Here, (6) means that the pair $(t, t + T)$ generates the same key, (7) implies that these T keys are mutually independent, (8) says that the keys are close to be uniformly generated, and (9) 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 a communication strategy $\mathbf{F}$ such that conditions (6)–(9) are satisfied and

$$R_t = \frac{1}{n} H(K_t), \quad t = 1, \cdots, T, \quad (10)$$

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

$$C_{sum} = \sup \sum_{t=1}^{T} R_t. \quad (11)$$

Remark 1: We note that in the problem formulation, each key is not required to be kept secret from other pairs. For

example, K_1 is not required to be kept secret from nodes 2 and $T+2$. As will be discussed in Section IV-C, our algorithm can be easily modified to take this constraint into consideration.

III. Network-Coding Based Single Group Key Generation

In this section, we consider the group key generation for a set of users A. The proposed scheme has two steps: 1) graph construction via local key generation; 2) key propagation using network coding. Our algorithm is based on a simple observation that the group key generation problem is closely related a multicast over an undirected network problem. This observation allows us to design a network coding based key generation scheme that outperforms the routing-based key generation scheme proposed in [16]. In addition, as will be clear in the sequel, finding the largest achievable rate using our scheme is a LP problem, while finding the largest achievable rate using the scheme in [16] is a NP-complete problem.

We describe our key generation algorithm in Algorithm 1 shown in the right column. We omit the details of the proof that the key generation approach specified in Algorithm 1 satisfies conditions (2)–(4), as it is similar to the proof for more sophisticated model considered in Section IV.

Remark 2: The key generation approach proposed in [16] also has two steps. The first step, namely graph construction, is similar to ours with a small difference. More specifically, in [16] the graph constructed is a multigraph with the number of edges (unit capacity) between i and j being $n(I(X_{ij};X_{ji})-\epsilon)$. The second step is different from ours. In [16], one finds edge disjoint Steiner trees. For each Steiner tree, the nodes in the set A generate one bit of group key via a simple algorithm. Although the algorithm is different from routing, it can be shown that it is equivalent to routing. In [16], it was shown that when $|A|=2$ or $|A|=m$, the routing-based scheme is optimal. However, for other cases, it is well-known that for a single source multicast, network coding outperforms routing. Hence, our scheme can achieve a larger key rate that of [16]. Here we give such an example. In this example, we have $m=7$, $A=\{1,2,3,4\}$, and the correlated random variables at the sources are

$$\tilde{X}_1=\{\phi,\phi,\phi,X_{15},X_{16},X_{17}\}, \tag{14}$$
$$\tilde{X}_2=\{\phi,\phi,\phi,X_{25},X_{26},\phi\}, \tag{15}$$
$$\tilde{X}_3=\{\phi,\phi,\phi,\phi,X_{36},X_{37}\}, \tag{16}$$
$$\tilde{X}_4=\{\phi,\phi,\phi,X_{45},\phi,X_{47}\}, \tag{17}$$
$$\tilde{X}_5=\{X_{51},X_{52},\phi,X_{54},\phi,\phi\}, \tag{18}$$
$$\tilde{X}_6=\{X_{61},X_{62},X_{63},\phi,\phi,\phi\}, \tag{19}$$
$$\tilde{X}_7=\{X_{71},\phi,X_{73},X_{74},\phi,\phi\}, \tag{20}$$

in which ϕ denotes no correlated component and for those components with non-zero correlation $I(X_{ij};X_{ji})=1$. Then the graph constructed using the local key generation step is shown in Figure 2. In the graph, the capacity of each edge is n. Using our approach, the users in A can generate $2n$ bits of keys, hence achieving a key rate of 2 bits per source observation. To achieve this, node 1 randomly generates

Algorithm 1 Network Coding Based Single Group Key Generation

- Step 1: Graph construction via local key establishment: Construct an undirected 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 $\mathcal{M}$. 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_{ij}^n,X_{ji}^n) [3]. 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. For any $\epsilon_1>0$, there exists a scheme [3] such that
$$\Pr\{K_{ij}\neq K_{ji}\}\leq\epsilon_1. \tag{12}$$
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). For any $\epsilon_2>0$, there exists a scheme [3], [6] such that
$$I(K_{ij};F_{ij})\leq\epsilon_2. \tag{13}$$
- Step 2: Key propagation via network coding: Based on the topology of the undirected graph constructed in the step 1, construct a network code [19] that achieves the largest multicast capacity from node 1 to all other users in set A using this network.[2] Let nR be the multicast capacity achieved in this network. Node 1 randomly generates a key K from the set $\{1,\cdots,2^{nR}\}$ using a uniform distribution. Node 1 then multicasts this key to other nodes in the set A using the optimal network coding identified above. For each hop $i\rightarrow j$, users i and j use the pairwise local key K_{ij} to encrypt and decrypt the codeword passed through this hop.

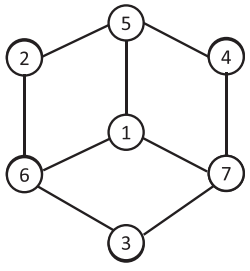

Fig. 2. The graph constructed from correlated sources after the local key generation step.

K with $2n$ bits, and divide it to two parts K_1^a and K_1^b each having n bits. Node 1 then multicasts K to the remaining

[2]The basic idea of constructing a network coding for an undirected graph in [19] has two steps: 1) Assign directions to each edge and construct the corresponding network code for the directed graph using the approach in [20]; 2) Find the best direction assignment that achieves the largest multicast throughput. [19] shows that finding the best orientation is a LP problem, and hence the complexity is low.

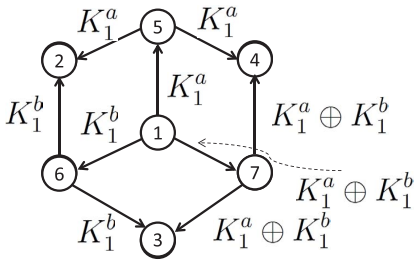

Fig. 3. The corresponding network coding to achieve the key rate of 2.

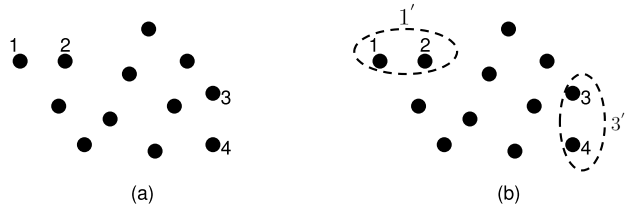

Fig. 4. (a) The original network, (b) a new model constructed from the original network.

nodes in A using the network code shown in Figure 3. Messages passing through each hop will be protected by the corresponding local key using the one-time pad. For example, the message $K_1^a \oplus K_1^b$ from node 7 to node 4 will be encrypted and decrypted using the local key K_{47}, which is n bits long, generated from the correlated observation (X_{47}, X_{74}). If only routing is used, using the result in [21], it can be shown that the key rate can only be 1.8 bits per source observation. Although the approach of [16] can achieve the optimal rates in some cases, our scheme outperforms the approach of [16] in this example.

Remark 3: Another advantage of our approach is its low complexity. Finding the largest achievable rate using the approach in [16] is equivalent to the Steiner tree packing [22], which is NP-complete. On the other hand, finding the largest achievable rate using our approach is a LP problem, as finding the capacity achieving network coding scheme in [21] is a LP problem, which can be solved efficiently.

Remark 4: In the second step of Algorithm 1, we construct a network coding by viewing node 1 as the information source. That is in our algorithm, node 1 generates the random key K, then multicasts this key to other nodes in A. As shown in [21], for the network coding over an undirected network, the multicast capacity remains the same no matter which node is chosen as the information source. Hence, in Algorithm 1, the key can be actually generated at any node in A, and be multicasted to other users in A. The achievable key rate will remain the same.

Remark 5: It is not clear whether or not our scheme achieves the key capacity characterized in [6].

IV. Generating Multiple Keys

In this section, we consider the generalization of generating multiple independent keys. We will show that a low complexity scheme that is a combination of local point-to-point key generation and a LP achieves the whole capacity region for the case of generating two keys, and is a constant fraction away from the maximum sum rate for the case of multiple keys. We achieve this by converting the key generation problem to a multi-commodity flow over a network problem, and exploit various results in graph theory [17].

A. 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 $\mathcal{M}$ and $B^c = \mathcal{M} \setminus B$. We say that a user pair (i, j) crosses $\{B, B^c\}$ if either 1) $i \in B$ and $j \in B^c$ or 2) $i \in B^c$ and $j \in B$. We first provide an outer bound on the key rate region.

Theorem 6: A rate pair (R_1, R_2) is achievable only if the following conditions are satisfied:

$$R_1 \leq \min_{B_1: 1 \in B_1, 3 \in B_1^c} \sum_{(i,j): i \in B_1, j \in B_1^c} I(X_{ij}; X_{ji}), \tag{21}$$

$$R_2 \leq \min_{B_2: 2 \in B_2, 4 \in B_2^c} \sum_{(i,j): i \in B_2, j \in B_2^c} I(X_{ij}; X_{ji}), \tag{22}$$

$$R_1 + R_2 \leq \min_{B_3: (1,2) \in B_3, (3,4) \in B_3^c} \sum_{(i,j): i \in B_3, j \in B_3^c} I(X_{ij}; X_{ji}), \tag{23}$$

$$R_1 + R_2 \leq \min_{B_4: (1,4) \in B_4, (2,3) \in B_4^c} \sum_{(i,j): i \in B_4, j \in B_4^c} I(X_{ij}; X_{ji}). \tag{24}$$

Proof: The basic idea of the proof is to construct a genie-aid model and show that the capacity region of this genie-aid model is upper bounded by (21)–(24).

The bound in (21) is an upper-bound on the key rate if we are required to generate only a single key K_1 [8], [16]. This obviously also serves as an upper-bound on the key rate of K_1 if we need to generate an additional key K_2. The bound in (22) can be obtained in a similar manner.

To derive the bound in (24), we consider a genie-aided model that is constructed from the original model. First, we create a super-terminal $1'$ by combining the observations at terminals $(1, 2)$. Similarly, we create a super-terminal $3'$ by combining the observations at terminals $(3, 4)$. In this newly constructed model, we are required to generate only one key for the pair of super-terminal $(1', 3')$. This is shown in Fig. 4. Now, for node $1'$, the observations are the n i.i.d. repetitions of

$$\begin{aligned} \tilde{Y}_{1'} &= \{Y_{1'3'}, Y_{1'5}, Y_{1'6}, \cdots, Y_{1'm}\} \\ &= \{(X_{13}, X_{14}, X_{23}, X_{24}), (X_{15}, X_{25}), \cdots, (X_{1m}, X_{2m})\}, \end{aligned} \tag{25}$$

the observations at node $3'$ are i.i.d. repetitions of

$$\begin{aligned} \tilde{Y}_{3'} &= \{Y_{3'1'}, Y_{3'5}, Y_{3'6}, \cdots, Y_{3'm}\} \\ &= \{(X_{31}, X_{41}, X_{32}, X_{42}), (X_{35}, X_{45}), \cdots, (X_{3m}, X_{4m})\}. \end{aligned} \tag{26}$$

For all other nodes $i \geq 5$, the observations are i.i.d. repetitions of

$$\begin{aligned}\tilde{Y}_i &= \{Y_{i1'}, Y_{i3'}, Y_{i5}, Y_{i6}, \cdots, Y_{im}\} \\ &= \{(X_{i1}, X_{i2}), (X_{i3}, X_{i4}), X_{i5}, X_{i6}, \cdots, X_{im}\}. \end{aligned} \quad (27)$$

Using the pair-wise independent nature of (X_{ij}, X_{ji}), $1 \leq i < j \leq m$, one can verify that random variables $\{(Y_{ij}, Y_{ji}), i, j \in \{1', 3', 5, \cdots, m\}\}$ are mutually independent. Now, for this new model, we need to generate a single key $K_{1'3'}$ with rate $R_{1',3'}$ for these two super-terminals $(1', 3')$. Since the observations at the terminals still possess the pair-wise independent structure, we have

$$R_{1',3'} \leq \min_{B_3: 1' \in B_3, 3' \in B_3^c} \sum_{i,j \in \{1',3',5,\cdots,m\}, (i,j), i \in B_3, j \in B_3^c} I(Y_{ij}; Y_{ji}). \quad (28)$$

It is easy to see that any scheme that generate keys (K_1, K_2) for the original model can be used to generate a key $K_{1',3'}$ with rate $R_{1',3'} = R_1 + R_2$. Hence, we have

$$R_1 + R_2 \leq \min_{B_3: 1' \in B_3, 3' \in B_3^c} \sum_{i,j \in \{1',3',5,\cdots,m\}, (i,j), i \in B_3, j \in B_3^c} I(Y_{ij}; Y_{ji}). \quad (29)$$

In the following, we simplify the right side of (29).

1) If $i, j \in \{5, \cdots, m\}$, we have

$$I(Y_{ij}; Y_{ji}) = I(X_{ij}; X_{ji}). \quad (30)$$

2) If $i = 1'$ and $j \in \{5, \cdots, m\}$, we have

$$\begin{aligned} I(Y_{1'j}; Y_{j1'}) &= I(X_{1j}, X_{2j}; X_{j1}, X_{j2}) && (31) \\ &= I(X_{1j}; X_{j1}) + I(X_{2j}; X_{j2}). && (32) \end{aligned}$$

3) If $i \in \{5, \cdots, m\}$ and $j = 3'$, we have

$$\begin{aligned} I(Y_{i3'}; Y_{3'i}) &= I(X_{i3}, X_{i4}; X_{3i}, X_{4i}) \\ &= I(X_{i3}; X_{3i}) + I(X_{i4}; X_{4i}). && (33) \end{aligned}$$

4) $i = 1'$ and $j = 3'$, we have

$$\begin{aligned} &I(Y_{1'3'}; Y_{3'1'}) \\ &\quad = I(X_{13}, X_{14}, X_{23}, X_{24}; X_{31}, X_{41}, X_{32}, X_{42}) \\ &\quad = I(X_{13}; X_{31}) + I(X_{14}; X_{41}) \\ &\qquad + I(X_{23}; X_{32}) + I(X_{24}; X_{42}). && (34) \end{aligned}$$

Hence, (29) is the same as

$$R_1 + R_2 \leq \min_{B_3: (1,2) \in B_3, (3,4) \in B_3^c} \sum_{(i,j), i \in B_3, j \in B_3^c} I(X_{ij}; X_{ji}), \quad (35)$$

which proves (23). (24) can be proved in a similar manner by combining terminals 1 and 4, and combining terminals 2 and 3. ■

In Algorithm 2 shown in the right column, 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.

In Fig. 5, we show an example of a network consisting of 5 nodes. Several routes are shown in the figure.

Algorithm 2 Generating Two Keys for Two Pairs

- Step 1: Graph Construction: Construct a graph $G_n(V, E)$ using the same approach as in Algorithm 1.
- Step 2: Key Propagation: Nodes 1 and 2 independently and randomly generate keys K_1 and K_2 from sets $\{1, \cdots, 2^{nR_1}\}$ and $\{1, \cdots, 2^{nR_2}\}$ 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 $\mathcal{P}_l^1 = (1, i_{l,2}, i_{l,3}, \cdots, 3)$ be the l^{th} route between node 1 and node 3, and Q_l^1 be the total number of hops in this route.[3] Node 1 divides key K_1 into L_1 non-overlapping parts $(K_1^1, K_2^1, \cdots, K_{L_1}^1)$, each having length W_l bits, and sends K_l^1 through the l^{th} route. Hence, we have total L_1 routes for key 1. In the q^{th} hop of the l^{th} route $(i_{l,q}, i_{l,q+1})$, node $i_{l,q}$ encrypts K_l^1 using W_l bits of the local key $K_{i_{l,q}, i_{l,q+1}}$. We use $K_{i_{l,q}, i_{l,q+1}}^{1,l}$ 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 $K_l^1 \oplus K_{i_{l,q}, i_{l,q+1}}^{1,l}$ over the public channel. After that, node $i_{l,q+1}$ decrypts K_l^1 using the same part of the local key $K_{i_{l,q}, i_{l,q+1}}$, namely $K_{i_{l,q}, i_{l,q+1}}^{1,l}$. After that, the node pair $(i_{l,q}, i_{l,q+1})$ will discard $K_{i_{l,q}, i_{l,q+1}}^{1,l}$, i.e., $K_{i_{l,q}, i_{l,q+1}}^{1,l}$, which will not be used again.[4] 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_l^2 hops.

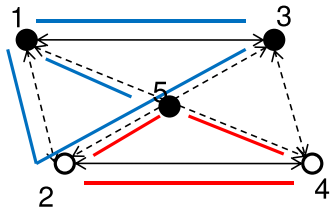

Fig. 5. An example of routes.

In this case, node 1 randomly generates a key K_1 and divides it into three parts (K_1^1, K_2^1, K_3^1), 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_1^2, K_2^2), which will be sent over routes $(2, 4), (2, 5, 4)$ to node 4 respectively. When K_1^1 is sent over the route $(1, 3)$, node 1 will send $K_1^1 \oplus K_{1,3}$. The node pair $(2, 5)$ will divide its local key $K_{2,5}$ into two parts, one is used to encrypt the third part of K_1, namely K_3^1, the other one is used to encrypt the second part of key K_2, namely K_2^2. 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 7: The key generation approach specified in Algorithm 2 satisfies conditions (6)–(9).

[3]Details on how to find the best routes will be discussed later.

[4]This will guarantee that the total amount of key information of node 1 and node 2 passing through each link will not be larger than the capacity of the corresponding link.

Proof: We note that (7) and (8) are satisfied since both K_1 and K_2 are independently generated using the uniform distribution. It is also easy to show that the error probability requirement is satisfied. The main challenging part is the key leakage analysis.

Three types of information have been exchanged over the public channel: 1) $\{F_{ij}, 1 \le i < j \le m\}$ that are used to establish local keys (in the following proof, to simplify notation, we use $\{F_{ij}\}$ to denote $\{F_{ij}, 1 \le i < j \le m\}$); 2) $\{K_l^1 \oplus K_{i_{l,q},i_{l,q+1}}^{1,l}, 1 \le l \le L_1, 1 \le q \le Q_l^1\}$ that are used to route $K_l^1, 1 \le l \le L_1$ from node 1 to node 3 (similarly, in the proof, to simplify the notation, we will use $\{K_l^1 \oplus K_{i_{l,q},i_{l,q+1}}^{1,l}\}$); and 3) $\{K_l^2 \oplus K_{i_{l,q},i_{l,q+1}}^{2,l}, 1 \le l \le L_2, 1 \le q \le Q_l^2\}$ that are used to route $K_l^2, 1 \le l \le L_2$ from node 2 to node 4 (again, in the proof, we will use $\{K_l^2 \oplus K_{i_{l,q},i_{l,q+1}}^{2,l}\}$). We have

$$\begin{aligned}
&I(K_1, K_2; \{F_{ij}\}, \{K_l^1 \oplus K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_l^2 \oplus K_{i_{l,q},i_{l,q+1}}^{2,l}\}) \\
&= I(K_1, K_2; \{F_{ij}\}) \\
&\quad + I(K_1, K_2; \{K_l^1 \oplus K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_l^2 \oplus K_{i_{l,q},i_{l,q+1}}^{2,l}\} | \{F_{ij}\}) \\
&= H(\{K_l^1 \oplus K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_l^2 \oplus K_{i_{l,q},i_{l,q+1}}^{2,l}\} | \{F_{ij}\}) \\
&\quad - H(\{K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_{i_{l,q},i_{l,q+1}}^{2,l}\} | \{F_{ij}\}) \\
&\le H(\{K_l^1 \oplus K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_l^2 \oplus K_{i_{l,q},i_{l,q+1}}^{2,l}\}) \\
&\quad - H(\{K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_{i_{l,q},i_{l,q+1}}^{2,l}\} | \{F_{ij}\}) \\
&\overset{(a)}{\le} H(\{K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_{i_{l,q},i_{l,q+1}}^{2,l}\}) \\
&\quad + (\max\{Q_l^1\} L_1 + \max\{Q_l^2\} L_2)\epsilon \\
&\quad - H(\{K_{i_{l,q},i_{l,q+1}}^{1,l}\}, \{K_{i_{l,q},i_{l,q+1}}^{2,l}\} | \{F_{ij}\}) \\
&\le (\max\{Q_l^1\} L_1 + \max\{Q_l^2\} L_2)(\epsilon + \epsilon_2), \qquad (36)
\end{aligned}$$

where we get (a) by repeatedly using the following inequality. In this inequality, there are four independent variables $Z_i, i = 1, \cdots, 4$ taking values from alphabet sets $\mathcal{Z}_i$, $i = 1, \cdots, 4$ respectively, then if $|\mathcal{Z}_1| \le |\mathcal{Z}_2|$, $|\mathcal{Z}_3| \le |\mathcal{Z}_4|$, $H(Z_2) \ge \log(|\mathcal{Z}_2|) - \epsilon$, and $H(Z_4) \ge \log(|\mathcal{Z}_4|) - \epsilon$, we have

$$\begin{aligned}
&H(Z_1 \oplus Z_2, Z_3 \oplus Z_4) \\
&\quad \le H(Z_1 \oplus Z_2) + H(Z_3 \oplus Z_4) \\
&\quad \le \log(|\mathcal{Z}_2|) + \log(|\mathcal{Z}_4|) \\
&\quad \le H(Z_2) + \epsilon + H(Z_4) + \epsilon \\
&\quad = H(Z_2, Z_4) + 2\epsilon. \qquad (37)
\end{aligned}$$

The bound in (36) can be made arbitrarily small as n increases. This implies that Eve learns negligible amount of information about the generated keys (K_1, K_2) from the public discussion and subsequent key routing process. ∎

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 $G_n(V, E)$ [17]. 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 6.

Theorem 8: The scheme in Algorithm 2 achieves the upper bound specified in Theorem 6 and hence is optimal.

Proof: The proof relies on the Max Bi-flows Min-Cut Theorem in graph theory establish in [23]. 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 [23], we know that as long as

$$f(1,3) \le \min_{1 \in B_1, 3 \in B_1^c} \sum_{(i,j): i \in B_1, j \in B_1^c} e_{ij}, \qquad (38)$$

$$f(2,4) \le \min_{2 \in B_2, 4 \in B_2^c} \sum_{(i,j): i \in B_2, j \in B_2^c} e_{ij}, \qquad (39)$$

$$f(1,3) + f(2,4) \le \min_{(1,3) \in B_3, (2,4) \in B_3^c} \sum_{(i,j): i \in B_3, j \in B_3^c} e_{ij}, \qquad (40)$$

$$f(1,3) + f(2,4) \le \min_{(1,4) \in B_4, (2,3) \in B_4^c} \sum_{(i,j): i \in B_4, j \in B_4^c} e_{ij}, \qquad (41)$$

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

$$f(1,3) = nR_1, \qquad (42)$$

$$f(2,4) = nR_2, \qquad (43)$$

$$e_{ij} = n(I(X_{ij}, X_{ji}) - \epsilon), \qquad (44)$$

into (38)–(41), we know that the secure routing-based key propagation approach achieves the outer bound established in Theorem 6. ∎

One can use the cycle flow method proposed in [23] 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.

B. General Case

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

The secure routing-based key propagation scheme discussed in Section IV-A can be used in this general scenario. In particular, we again construct a undirected graph $G_n(V, E)$ with V being the same as $\mathcal{M}$ and E being the set of edge of capacity $e_{ij} = n(I(X_{ij}; X_{ji}) - \epsilon)$. Node $t, 1 \le t \le T$ then randomly generates a key K_t using a uniform distribution from the set $\{1, \cdots, 2^{nR_t}\}$. Node t further divides K_t into non-overlapping L_t parts $(K_1^t, \cdots, K_{L_t}^t)$, where each part is sent over a route from node t to node $t+T$. During the routing, each key part K_i^t is encrypted and decrypted using the local

keys established from the pair-wise correlated observations. Following the same steps in the proof of Theorem 7, 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 an 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 fractional multi-commodity flows,[5] which has been extensively studied in graph theory. In particular, one can formulate a LP to characterize C_r. In order to write the largest achievable rate in a concise manner, we add special edges $(T+t, t)^s$ with infinite capacity to the set E that allows only commodity of type t to flow from user $T+t$ to user t. We use $\tilde{E} = E \cup \{(T+t, t)^s, t = 1, \cdots, T\}$ to denote this enhanced set of edges. Then, C_r is the value of the following LP [17]

$$\max \quad \frac{1}{n}\sum_{t=1}^{T} f^t_{(T+t,t)^s} \tag{45}$$

$$\text{s.t.} \quad \sum_{(j,i)\in\tilde{E}} f^t_{ji} - \sum_{(i,j)\in\tilde{E}} f^t_{ij} = 0, \ \forall i \in V, \ \forall t \in [1, \cdots, T], \tag{46}$$

$$\sum_{t=1}^{T} f^t_{ij} + \sum_{t=1}^{T} f^t_{ji} \le e_{ij}, \quad \forall (i, j) \in E, \tag{47}$$

$$f^t_{ij} \ge 0, \forall (i, j) \in \tilde{E}, \quad \forall t \in [1, \cdots, T], \tag{48}$$

in which f^t_{ij} is the amount of key information of user pair $(t, t+T)$ that passed through from i to j. (46) implies that the total flow of each commodity into node i is the same as the total flow out it, and (47) implies that total amount of information flow through edge (i, j) must be smaller than $e_{i,j}$. Since it is a LP problem, efficient algorithms to find the best routes and the corresponding largest achievable rate exist.

We now develop an upper bound for the sum of key rates for any key generation protocols (not necessarily limited to the two-step approach proposed in this paper). We will use a graph $G^*_n(V, E)$ that is the same as $G_n(V, E)$ constructed above with a modification that the link capacity $e_{ij} = nI(X_{ij}; X_{ji})$. 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 $t = 1, \cdots, T$. Equivalently, a set E' is a multicut if for all $t = 1, \cdots, T$, 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 $V_1, V_2, \cdots, V_U$ such that for all $t = 1, \cdots, T$, node t and node $t+T$ are in two different subsets. It is easy to see that $U \le m$. For each node set V_u with $u = 1, \cdots, U$, 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 6 illustrates a multicut

[5]Although the number of bits passed through the network should be integer, rounding an optimal fractional solution to an integer solution will not affect the rate.

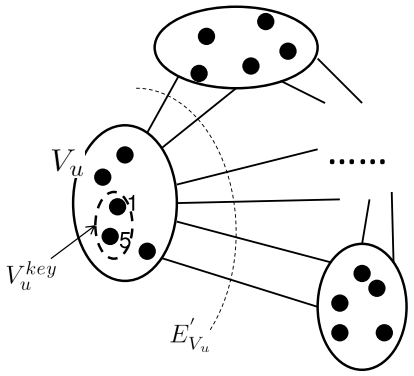

Fig. 6. An example of multicut and related definitions.

and the associated definitions. The value of a multicut E' is defined as

$$C_{E'} = \sum_{(i,j)\in E'} e_{ij} = \sum_{(i,j)\in E'} nI(X_{ij}; X_{ji}). \tag{49}$$

We have the following upper bound on the sum rate of key rates for any key generation protocol.

Theorem 9: For any key generation protocol (not necessarily limited to the two-step approach discussed in the paper), the sum capacity is upper-bounded by the following

$$C_{sum} \le \frac{1}{n}\min_{E'} C_{E'} = \min_{E'} \sum_{(i,j)\in E'} I(X_{ij}; X_{ji}). \tag{50}$$

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

$$E'_{V_u} = \{(i, j) : i \in V_u, j \in V_u^c\}. \tag{51}$$

For any V_u of a given multicut E', we consider a genie-aided model created as following. We first create a super-node V_u^* by combining all observations at nodes in V_u^{key} and another super-node $V_u^{*,c}$ by combining all observations at nodes in $V_u^{key,c}$. 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 6, we know that

$$nR_{V_u} \le \sum_{(i,j)\in E'_{V_u}} nI(X_{ij}; X_{ji}). \tag{52}$$

Clearly, any key generation protocol for the original model with key rate R_i for $i \in V_u^{key}$ can be used to generate a key for this two super nodes in this genie aided model[6]. Hence

$$n\sum_{i\in V_u^{key}} R_i \le nR_{V_u} \le \sum_{(i,j)\in E'_{V_u}} nI(X_{ij}; X_{ji}). \tag{53}$$

[6]Here for notation convenience, we allow i to be in the range of 1 and $2T$ with the understanding that $R_i = R_{i-T}$ if $i > T$.

One can repeat the same steps as above for each V_u, $1 \le u \le U$, and (53) is true for each u.

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

$$n\sum_{u=1}^{U}\sum_{i\in V_u^{key}} R_i \le \sum_{u=1}^{U}\sum_{(i,j)\in E'_{V_u}} nI(X_{ij};X_{ji}). \tag{54}$$

Noting that

$$\sum_{u=1}^{U}\sum_{i\in V_u^{key}} R_i = 2\sum_{i=1}^{T} R_i, \tag{55}$$

$$\sum_{u=1}^{U}\sum_{(i,j)\in E'_{V_u}} nI(X_i;X_j) = 2C_{E'}, \tag{56}$$

we have

$$n\sum_{i=1}^{T} R_i \le C_{E'}. \tag{57}$$

Since (57) is true for any multicut E', we have

$$C_{sum} \le \frac{1}{n}\min_{E'} C_{E'} = \min_{E'}\sum_{(i,j)\in E'} I(X_{ij};X_{ji}). \tag{58}$$

■

The following theorem characterizes the relationship between the sum rate C_r achieved using our routing-based approach, which is characterized in (45), to that of the upper bound derived in Theorem 9.

Theorem 10:

$$C_r \ge C_{sum}/O(\log T), \tag{59}$$

where C_{sum} and $O(\log T)$ have the same base of log.

Proof: The proof is an application of a result in graph theory that characterizes the relationship between max sum flow and min multi-cut [24] of a graph. More specifically, for a graph $G_n(V, E)$, using [24, Th. 5.1], we have

$$nC_r \ge \frac{1}{O(\log T)}\min_{E'} C_{E'}. \tag{60}$$

Coupled with (50), we have the desired result. ■

Remark 11: Unlike the single group key generation scenario, it is not clear whether or not network coding will improve the achievable sum key rate in this multiple-key generation setup. After the graph construction, the problem studied in this section is essentially a multiple unicasts over an undirected network problem. It has been conjectured that network coding does not bring benefit for multiple unicasts over an undirected network [25]. However, until now, this conjecture has not been proved or disproved.

C. Mutual Privacy

The proposed scheme can be easily modified to satisfy additional constraints. One such constraint is that the generated keys should also be kept secret from other user pairs. To satisfy this constraint, we need only to avoid routing the traffic of user t through nodes $[1,\cdots,2T]\setminus\{t, T+t\}$. Finding the largest achievable sum rate using the proposed scheme is again a LP with additional constraint. In particular, the maximum achievable rate with this additional privacy constraint using our scheme is the solution of the following LP:

$$\max \quad \frac{1}{n}\sum_{t=1}^{T} f^t_{(T+t,t)^s} \tag{61}$$

$$\text{s.t.} \quad \sum_{(j,i)\in\tilde{E}} f^t_{ji} - \sum_{(i,j)\in\tilde{E}} f^t_{ij} = 0, \tag{62}$$

$$\forall i \in V, \forall t \in [1,\cdots,T], \tag{63}$$

$$\sum_{t=1}^{T} f^t_{ij} + \sum_{t=1}^{T} f^t_{ji} \le e_{ij}, \tag{64}$$

$$\forall (i,j) \in E, \tag{65}$$

$$f^t_{ij} \ge 0, \quad \forall (i,j)\in\tilde{E}, \quad \forall t \in [1,\cdots,T], \tag{66}$$

$$f^t_{ij} = 0, \text{ if } i \text{ or } j \in [1,\cdots,2T] \tag{67}$$

$$\text{and at least one of } i \text{ and } j \text{ is not } t \text{ and } t+T. \tag{68}$$

Here, the additional constraint (68) implies that the key information of user pair $(t, t+T)$ will not be pass through other users that are required to generate keys.

V. Conclusion

We have considered two scenarios for key generation under PIN model. In the first scenario, in which one is required to generate a group key, we have proposed a network coding based approach. The approach has a low complexity and has a better performance than the existing approach. In the second scenario, we have considered the problem of simultaneously generating multiple keys. 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. For the general case of generating more than two keys, we have also shown that the sum rate of the proposed scheme is larger than an upper bound characterized in this paper divided by a constant. Furthermore, finding the largest achievable sum rate using our scheme is a LP problem. The proposed scheme can also be easily modified to take additional constraints into consideration.

References

[1] L. Lai and S.-W. Ho, "Simultaneously generating multiple keys and multi-commodity flow in networks," in *Proc. IEEE Inf. Theory Workshop*, Lausanne, Switzerland, Sep. 2012, pp. 627–631.

[2] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[3] R. Ahlswede and I. Csiszár, "Common randomness in information theory and cryptography. I. Secret sharing," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1121–1132, Jul. 1993.

[4] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1495–1507, Apr. 2006.

[5] C. Ye and P. Narayan, "Secret key and private key constructions for simple multiterminal source models," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 639–651, Feb. 2012.

[6] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3047–3061, Dec. 2004.

[7] C. Chan and L. Zheng, "Mutual dependence for secret key agreement," in *Proc. 44th Annu. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2010, pp. 1–6.
[8] C. Ye and A. Reznik, "Group secret key generation algorithms," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 2596–2600.
[9] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 364–375, Sep. 2007.
[10] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 240–254, Jun. 2010.
[11] L. Lai, Y. Liang, and H. V. Poor, "A unified framework for key agreement over wireless fading channels," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 480–490, Apr. 2012.
[12] T.-H. Chou, A. M. Sayeed, and S. C. Draper, "Impact of channel sparsity and correlated eavesdropping on secret key generation from multipath channel randomness," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, Jun. 2010, pp. 2518–2522.
[13] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radiotelephathy: Extracting a secret key from an unauthenticated wireless channel," in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, San Francisco, CA, USA, Sep. 2008, pp. 128–139.
[14] A. Khisti, "Interactive secret key generation over reciprocal fading channels," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2012, pp. 1374–1381.
[15] M. J. Siavoshani, C. Fragouli, S. Diggavi, U. Pulleti, and K. Argyraki, "Group secret key generation over broadcast erasure channels," in *Proc. Conf. Rec. 44th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2010, pp. 719–723.
[16] S. Nitinawarat, C. Ye, A. Barg, P. Narayan, and A. Reznik, "Secret key generation for a pairwise independent network model," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6482–6489, Dec. 2010.
[17] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[18] C. Chan and L. Zheng, "Network coding for secret key agreement," *IEEE Trans. Inf. Theory*. Submitted.
[19] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2467–2485, Jun. 2006.
[20] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
[21] Z. Li, B. Li, and L. C. Lau, "A constant bound on throughput improvement of multicast network coding in undirected networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1016–1026, Mar. 2009.
[22] S. Chen, O. Gunluk, and B. Yener, "The multicast packing problem," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 311–318, Jun. 2000.
[23] T. C. Hu, "Multi-commodity network flows," *Operat. Res.*, vol. 11, no. 3, pp. 344–360, May/Jun. 1963.
[24] N. Garg, V. V. Vazirani, and M. Yannakakis, "Approximate max-flow min-(multi)cut theorems and their applications," *SIAM J. Comput.*, vol. 25, no. 2, pp. 235–251, Apr. 1996.
[25] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Proc. 42nd Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2004, pp. 11–19.

Lifeng Lai (M'07) received the B.E. and M. E. degrees from Zhejiang University, Hangzhou, China in 2001 and 2004 respectively, and the PhD degree from The Ohio State University at Columbus, OH, in 2007.

He was a postdoctoral research associate at Princeton University from 2007 to 2009, and was an assistant professor at University of Arkansas, Little Rock from 2009 to 2012. Since Aug. 2012, he has been an assistant professor at Worcester Polytechnic Institute. His research interests include information theory, stochastic signal processing and their applications in wireless communications, security and other related areas.

Dr. Lai was a Distinguished University Fellow of the Ohio State University from 2004 to 2007. He is a co-recipient of the Best Paper Award from IEEE Global Communications Conference (Globecom) in 2008, the Best Paper Award from IEEE Conference on Communications (ICC) in 2011 and the Best Paper Award from IEEE Smart Grid Communications (SmartGridComm) in 2012. He received the National Science Foundation CAREER Award in 2011, and Northrop Young Researcher Award in 2012. He served as a Guest Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Special Issue on Signal Processing Techniques for Wireless Physical Layer Security. He is currently serving as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.

Siu-Wai Ho (S'05–M'07–SM'15) received the B.Eng., M.Phil., and Ph.D. degrees in information engineering from The Chinese University of Hong Kong in 2000, 2003, and 2006, respectively.

During 2006-2008, he was a Postdoctoral Research Fellow in the Department of Electrical Engineering, Princeton University, Princeton, NJ. Since 2009, he has been with the Institute for Telecommunications Research (ITR) in University of South Australia (UniSA), Adelaide, Australia, where he is now a senior research fellow. His current research interests include Shannon theory, visible light communications, informationtheoretic security, and biometric security systems.

Dr. Ho was a recipient of the Croucher Foundation Fellowship for 2006-2008, the 2008 Young Scientist Award from the Hong Kong Institution of Science, UniSA Research SA Fellowship for 2010-2013, and the Australian Research Council Australian Postdoctoral Fellowship for 2010-2013.