# SYNERGY: A Game-Theoretical Approach for Cooperative Key Generation in Wireless Networks

Jingchao Sun, Xu Chen, Jinxue Zhang, Yanchao Zhang, and Junshan Zhang
School of Electrical, Computer and Energy Engineering (ECEE)
Arizona State University, Tempe, Arizona, USA
{jcsun, xchen179, jxzhang, yczhang, junshan.zhang}@asu.edu

***Abstract*—This paper studies secret key establishment between two adjacent mobile nodes, which is crucial for securing emerging device-to-device (D2D) communication. As a promising method, cooperative key generation allows two mobile nodes to select some common neighbors as relays and directly extract a secret key from the wireless channels among them. A challenging issue that has been overlooked is that mobile nodes are often self-interested and reluctant to act as relays without adequate reward in return. We propose SYNERGY, a game-theoretical approach for stimulating cooperative key generation. The underlying idea of SYNERGY is to partition a group of mobile nodes into disjoint coalitions such that the nodes in each coalition fully collaborate on cooperative key generation. We formulate the group partitioning as a coalitional game and design centralized and also distributed protocols for obtaining the core solution to the game. The performance of SYNERGY is evaluated by extensive simulations.**

## I. Introduction

Device-to-device (D2D) communication is quickly emerging due to the ever-growing popularity of powerful mobile devices and also the rapid advance in D2D technologies [1]. In a typical D2D session, adjacent mobile devices can directly communicate without involving a base station. The competing technologies for establishing D2D connections include Bluetooth and WiFi-direct over the unlicensed band as well as LTE-A over the licensed band. D2D communication is promising to enhance spectrum efficiency and system throughput, enable efficient traffic offloading, improve energy efficiency and network coverage, and stimulate excitingly new services.

A key challenge for advancing D2D communication is to secure a D2D connection given the ease of malicious eavesdropping on wireless transmissions. One may think about encrypting and authenticating the content sent over a D2D connection based on a secret key shared between two mobile nodes. A conventional way for establishing secret keys depends on each node owning a public-key certificate, but it is unlikely to have a public-key certificate on every mobile node in the near future. Another traditional method is through a trusted third party which may not exist in most scenarios; even if there were one, heavily involving it may largely offset the benefits from autonomous D2D communication.

It is more promising to generate a secret key directly from the wireless channel between two mobile nodes. Specifically, according to the channel reciprocity theory, the channel responses between two wireless devices share some common randomness which is unavailable and also unpredictable to any eavesdropper more than one-half wavelength away from both devices. There have been some efforts (e.g., [2]–[5]) whereby two mobile nodes can extract a secret key from such common channel randomness. The resulting key is information-theoretically secure, and it can be generated on demand and updated dynamically in line with time-varying and location-dependent wireless channels [6]. In addition, there is no requirement for a trusted third party or prior trust relationship between two mobile nodes. This PHY (short for physical layer) approach is thus very suitable for secure D2D communication. The rate at which secret bits are generated from the wireless channel heavily depends on how fast the channel changes. In slowly changing wireless environments, the key generation rate may be very low. This practical limitation is widely reported [4], [6], [7] and may jeopardize the potential of PHY-based secret key generation in D2D scenarios with high security demand.

Cooperative key generation [6], [8] can be leveraged to improve the key generation rate of the PHY approach by incorporating additional randomness. The main idea is to explore some relay nodes in the vicinity of two target nodes and use the random channels associated with these relay nodes as additional random sources for secret key generation between the two target nodes. The efficacy of cooperative key generation is well analyzed and confirmed in various scenarios [6], [8]. The feasibility of this technique is, however, still questionable, as mobile nodes are self-interested in nature and typically reluctant to participate if they cannot get adequate benefit from the cooperation.

We propose SYNERGY, a game-theoretical approach for stimulating cooperative key generation in wireless networks. SYNERGY targets at a multi-hop D2D communication scenario in which every node wishes to establish a secret key with at least one neighbor via the PHY approach. The underlying idea of SYNERGY is to partition all the nodes involved into multiple disjoint coalitions. Every node in a coalition is strongly motivated to help other nodes in the same coalition establish secret keys to get help in return.

Our contributions can be summarized as follows. First, this work is the first to study incentive-aware cooperation key generation in wireless networks, to the best of our knowledge. Second, we formulate it as a coalitional game and devise an algorithm to find the core solution. Third, we propose centralized and distributed implementations for the core discovery algorithm. Finally, we show that SYNERGY is highly efficient and effective through extensive simulations.

In what follows, Section III outlines the background for

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

cooperative key generation in wireless networks. Section III gives the system and adversary models. Section IV presents a coalitional game formulation for incentive-aware cooperative key generation and the algorithm for obtaining the core solution. Section V introduces centralized and distributed implementations of SYNERGY and analyzes their performance. Section VI evaluates SYNERGY using simulations. Section VII briefs the related work. Section VIII concludes this paper.

## II. Background

For the sake of completeness, this section outlines the basics of PHY-based noncooperative and cooperative secret key generation.

### *A. PHY-based Noncooperative Key Generation*

Assume that two nodes Alice (A) and Bob (B) want to establish a shared secret key via the wireless channel between them in the presence of an eavesdropper Eve (E). Both Alice and Bob can transmit, while Eve only passively eavesdrops on wireless transmissions to avoid being detected.

Key generation starts by Alice sending a signal $X_A$. Then Bob and Eve will receive $Y_B = h_{AB}X_A + n_B$ and $Y_E = h_{AE}X_A + n_E$, respectively. Next, Bob transmits a signal $X_B$, and Alice and Eve will receive $Y_A = h_{BA}X_B + n_A$ and $Y_E = h_{BE}X_B + n_E$, respectively. Here $h_{AB}$, $h_{AE}$, $h_{BA}$, and $h_{BE}$ denote the channel gains from Alice to Bob, from Alice to Eve, from Bob to Alice, and from Bob to Eve, respectively; $n_A$, $n_B$, and $n_E$ are all commonly assumed to be zero-mean additive Gaussian noise with variance $\sigma^2$.

The wireless channel between Alice and Bob is assumed to be reciprocal, which means that $h_{AB} \cong h_{BA}$. In addition, assuming that Eve is more than one-half wavelength away from Alice and Bob, $h_{AB}$ and $h_{AE}$ are thus uncorrelated, so are $h_{BA}$ and $h_{BE}$. Also assume that the channel response is a Gaussian random variable with zero mean and variance $\sigma_1^2$. According to [9], the optimal key generation rate is

$$R_{A,B} = \frac{1}{T} I(\tilde{h}_{AB}; \tilde{h}_{BA}) = \frac{1}{2T} \log_2 \left(1 + \frac{\sigma_1^4 P^2 T^2}{4(\sigma^4 + \sigma^2\sigma_1^2 PT)}\right), \tag{1}$$

where $\tilde{h}_{AB}$ denotes Bob's estimate of $h_{AB}$, $\tilde{h}_{BA}$ denotes Alice's estimation of $h_{BA}$, $I(\tilde{h}_{AB}, \tilde{h}_{BA})$ denotes the mutual information [10] of $\tilde{h}_{AB}$ and $\tilde{h}_{BA}$, $T$ is the number of symbols during which the channel gains are fixed (i.e., coherence time), and $P$ is the transmission power of each node.

### *B. PHY-based Cooperative Key Generation*

Cooperative key generation [6], [8] via relay nodes is proposed to improve the key generation rate of the above noncooperative approach. The underlying idea is to explore the additional randomness brought by other nodes in the vicinity of Alice and Bob. Consider the example in Fig. 1, where Charlie (C) and Dave (D) are two common neighbors of Alice and Bob and thus can both serve as a relay. Cooperative key generation involving one relay, say Charlie, consists of two steps: channel estimation and key generation.

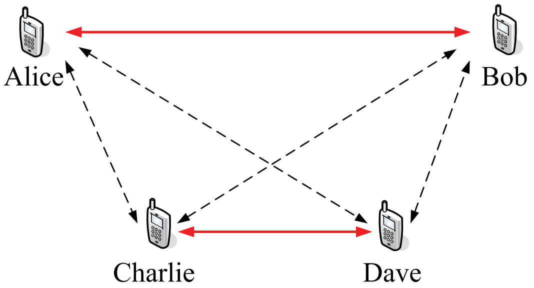


Fig. 1: PHY-based cooperative key generation. Dashed and solid lines both denote neighboring relationships, and a solid line additionally means that the two line ends (i.e., two peer nodes) want to establish a secret key.

**Channel Estimation**

1. Alice sends a known sequence $\mathbf{S}_A$, from which Bob and Charlie estimate the channel gains $h_{AB}$ and $h_{AC}$ as $\tilde{h}_{AB}$ and $\tilde{h}_{AC}$, respectively.
2. Bob sends a known sequence $\mathbf{S}_B$, from which Alice and Charlie estimate the channel gains $h_{BA}$ and $h_{BC}$ as $\tilde{h}_{BA}$ and $\tilde{h}_{BC}$, respectively.
3. Charlie sends a known sequence $\mathbf{S}_C$, from which Alice and Bob estimate the channel gains $h_{CA}$ and $h_{CB}$ as $\tilde{h}_{CA}$ and $\tilde{h}_{CB}$, respectively.

**Key Agreement**

1. Alice and Bob establish a secret key $K_{AB}$ based on $\tilde{h}_{AB}$ and $\tilde{h}_{BA}$. In addition, Alice and Charlie establish on a secret key $K_{AC}$ based on $\tilde{h}_{AC}$ and $\tilde{h}_{CA}$. Finally, Bob and Charlie establish a secret key $K_{BC}$ based on $\tilde{h}_{BC}$ and $\tilde{h}_{CB}$.
2. Charlie broadcasts $K_{AC} \oplus K_{BC}$, from which Alice and Bob each know both $K_{BC}$ and $K_{AC}$. If $K_{AC}$ is shorter than $K_{BC}$, Alice and Bob set the final secret key as $K_{AB} \parallel K_{AC}$ and $K_{AB} \parallel K_{BC}$ otherwise.

According to the result of [6], the optimal key generation rate of this cooperative approach is

$$R_{A,B}^{(C)} = \frac{1}{T}\Big\{ \min\{I(\tilde{h}_{CA}; \tilde{h}_{AC}), I(\tilde{h}_{CB}; \tilde{h}_{BC})\} + I(\tilde{h}_{AB}; \tilde{h}_{BA}) \Big\}. \tag{2}$$

Similarly, if both Charlie and Dave act as relays, the optimal key generation rate is given by [6]

$$R_{A,B}^{(C,D)} = \frac{1}{T}\Big\{ I(\tilde{h}_{AB}; \tilde{h}_{BA}) + \min\{I(\tilde{h}_{CA}; \tilde{h}_{AC}), I(\tilde{h}_{CB}; \tilde{h}_{BC})\} + \min\{I(\tilde{h}_{DA}; \tilde{h}_{AD}), I(\tilde{h}_{DB}; \tilde{h}_{BD})\} \Big\}. \tag{3}$$

We have two important remarks to make. First, the optimal key generation rates given above are only in information-theoretical sense. In practice, any two nodes involved have to generate a secret key from their channel estimates by following a few steps in sequel, including quantization, information reconciliation, and privacy amplification, as in [2]–[5]. So the real key generation rates are usually smaller. Second, it can be seen from our illustration above that the relay node(s) know partial information about the eventual secret key. If this is a concern, a more advanced and also complicated technique in [6] can be applied instead. Our proposed SYNERGY can work

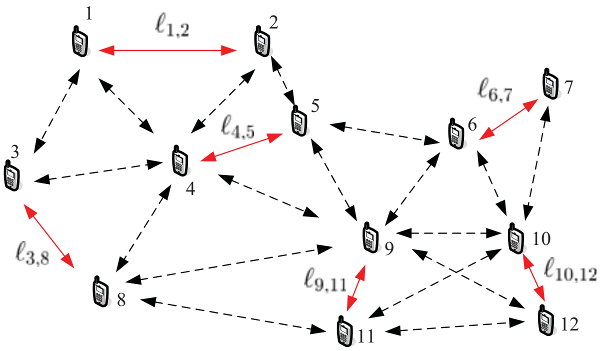


Fig. 2: An exemplary multi-hop D2D scenario, where dashed and solid lines both denote neighboring relationships, and a solid line additionally means that the two line ends (i.e., two peer nodes) want to establish a secret key.

with both techniques, but we focus on the basic technique above to facilitate the presentation.

## III. System and Adversary Models

We consider a multi-hop D2D scenario, in which every mobile node has at least one mobile device ready for D2D communication via Bluetooth, WiFi-direct, LTE-A, or other available D2D technologies. To enable analytical tractability, we assume that every node has the same transmission power and range. Mobile nodes are assumed to be *selfish* and *rational*. By selfish, we mean that every node will not act as a relay to help other nodes establish a secret key without a sound incentive. Our goal is to divide the nodes into disjoint coalitions, in which every node assists others establishing a secret key and also gets help from others in return. By rational, we mean that every node in a coalition faithfully follows the protocol operations and collaborates with others on key generation.

We consider the following adversary model commonly adopted for PHY-based key generation [2]–[6], [8]. Specifically, the adversary only passively eavesdrops on the wireless channel without actively jamming the channel. It is more than one-half length away from any two neighboring nodes trying to establish a secret key. Therefore, the adversary can only obtain the noisy versions of the wireless transmissions between two target nodes, so it cannot directly construct the secret key between them. For example, for wireless transmissions in the 2.4 GHz band, we only require the adversary to be more than 6.25 cm away from target nodes. This assumption is thus easily justifiable in practice. As in [6], [8], we assume that the nodes serving as relays for secret key generation do not cooperate with the adversary or other relays to obtain useful information about secret keys. There may be multiple eavesdroppers, which are assumed to be independent from each other. How to deal with collaborative eavesdroppers is still an open challenge.

## IV. SYNERGY: Cooperative Key Generation based on Social Reciprocity

As noted above, a key challenge for adopting cooperative key generation [6], [8] for D2D communication is the natural self-interest of mobile nodes: nobody wants to spend scarce system resources as a relay without getting adequate reward. We propose SYNERGY to tackle this open challenge based on the powerful theory of social reciprocity. The essential idea of SYNERGY is that a mobile node can be strongly motivated to act as a relay for other nodes if it could also get help in return to generate a secret key for itself. More specifically, SYNERGY partitions a set of mobile nodes into disjoint coalitions, each comprising some nodes acting as relays for others in cooperative key generation in order to improve their respective key generation rate. The main design challenge for SYNERGY lies in the partitioning rule for a given set of mobile nodes. In this section, we formulate this challenging issue as a coalitional game and describes an algorithm to find the core solution to the game.

### A. Notation and Terms

We consider $N$ mobile nodes denoted by $\mathcal{N} = \{1, \ldots, N\}$ and define the following terms to facilitate the illustration.

- **Peer**: Two nodes are said to be peers of each other iff they are physical neighbors and want to establish a secret key. The two nodes are called a *peer pair*.
- **Relay**: A relay of a peer pair refers to a node which is their common neighbor and helps them establish a secret key through cooperative key generation.
- **Contributor**: If either or both of two peers serves as a relay for another peer pair, we say that the first peer pair is a contributor to the second pair.

We assume that each node has one and only one peer in any SYNERGY session, which means that $N$ is even. Due to the limited transmission range of mobile nodes, we can view $\mathcal{N}$ as the vertex set of an undirected graph, where every edge corresponds to two neighboring nodes. Assume that every peer pair has at least one common neighbor as a candidate relay. Otherwise, the peer pair can only establish a secret key using noncooperative key generation and does not need to participate in SYNERGY operations. Let $\mathcal{C}_{i,j} \neq \phi$ denote the common neighbors of peers $i$ and $j$. We further assume that $i$ and $j$ can use no more than two relays (if any) from $\mathcal{C}_{i,j} \neq \phi$, and the two-relay case can only occur when the two relays themselves compose a peer pair. The extension of SYNERGY to more general cases is very challenging and left as future work. For simplicity, we also assume that every node acts as a relay at most once in a SYNERGY session.

As an example, we have $\mathcal{N} = \{1, \ldots, 12\}$, $\mathcal{C}_{1,2} = \{4\}$, $\mathcal{C}_{4,5} = \{2, 9\}$, $\mathcal{C}_{9,11} = \{8, 10, 12\}$in Fig. 2. Peers 9 and 11 can potentially use nodes 10 and 12 as two relays because nodes 10 and 12 also form a peer pair. In contrast, peers 4 and 5 can have at most one relay, either node 2 or 9.

### B. Coalitional Game Formulation

In game theory, a coalitional game refers to a game where a competition is between coalitions of players instead of between individual players [11]. It is thus a very natural tool for incentive-aware cooperative key generation.

Our coalitional game formulation relies on a special trick. We introduce a **virtual node** (denoted by $\ell_{i,j}$) for every peer pair $i$ and $j$, as shown in Fig. 2. Note that we have $\ell_{i,j} = \ell_{j,i}$. Now consider any other virtual node $\ell_{s,d}$ $(i \neq j \neq s \neq d)$. If either or both of $s$ and $d$ act as a relay for peers $i$ and $j$, we say that $\ell_{s,d}$ **contributes** to $\ell_{i,j}$. Since every peer pair can be assumed to have a common interest in improving their key generation rate, we can use the $N/2$ virtual nodes as the game players rather than the $N$ real nodes.

What is the most preferred contributor of every virtual node $\ell_{i,j}$, or equivalently the most preferred relay of peers $i$ and $j$? Recall that the optimal key generation rates with one relay and two relays are given in Eq. (2) and Eq. (3), respectively. To answer the preceding question, let $\mathcal{L}_{i,j}$ denote the set of potential contributors to $\ell_{i,j}$. For example, we have $\mathcal{L}_{4,5} = \{\ell_{1,2}, \ell_{9,11}\}$ in Fig. 2. Consider any virtual node in $\mathcal{L}_{i,j}$, say $\ell_{s,d}$. It contributes one potential relay to $\mathcal{L}_{i,j}$ if only one of $s$ and $d$ is a common neighbor of $i$ and $j$ and two potential relays if both $s$ and $d$ are a common neighbor of $i$ and $j$. Accordingly, we define the key-rate function of $\ell_{i,j}$ with regard to any potential contributor $\ell_{s,d} \in \mathcal{L}_{i,j}$ as

$$\hat{R}_{i,j}^{s,d} = \begin{cases} R_{i,j}^{(s)} & \forall s \in \mathcal{C}_{i,j}, d \notin \mathcal{C}_{i,j} \\ R_{i,j}^{(d)} & \forall s \notin \mathcal{C}_{i,j}, d \in \mathcal{C}_{i,j} \\ R_{i,j}^{(s,d)} & \forall s, d \in \mathcal{C}_{i,j}\ . \end{cases} \tag{4}$$

The most preferred virtual node or contributor of $\ell_{i,j}$ can then be defined as $r_{i,j}^{*} = \underset{\ell_{s,d} \in \mathcal{L}_{i,j}}{\operatorname{argmax}} \hat{R}_{i,j}^{s,d}$.

Based on the concepts above, we now formulate incentive-aware cooperative key generation as a coalitional game $\Omega = \langle \mathcal{L}, \mathcal{X}_{\mathcal{L}}, \Theta, (\succ_{i,j})_{\ell_{i,j} \in \mathcal{L}} \rangle$ as follows.

- **Players**: $\mathcal{L}$ denotes the set of game players consisting of all the $N/2$ virtual nodes.
- **Strategies**: $\mathcal{X}_{\mathcal{L}}$ denotes the set of feasible cooperation strategies (i.e., contributor selections) for all the players. We denote the contributor chosen by any player $\ell_{i,j} \in \mathcal{L}$ by $r_{i,j} \in \mathcal{L}$. It follows that $\mathcal{X}_{\mathcal{L}} = \{r_{i,j} | r_{i,j} \in \mathcal{L}, \forall \ell_{i,j} \in \mathcal{L}\}$.
- **Characteristic function**: Every virtual node in every coalition $\mathcal{S} \subseteq \mathcal{L}$ selects one and only one other virtual node in $\mathcal{S}$ as a contributor. In addition, every virtual node outside $\mathcal{S}$ cannot get an contributor from $\mathcal{S}$. Therefore, the characteristic function for every coalition $\mathcal{S} \subseteq \mathcal{L}$ can be denoted as $\Theta(\mathcal{S}) = \big\{ \{r_{i,j}\}_{\ell_{i,j} \in \mathcal{S}} = \{\ell_{i,j}\}_{\ell_{i,j} \in \mathcal{S}}, \{r_{i,j} = \ell_{i,j}\}_{\ell_{i,j} \in \mathcal{L} \setminus \mathcal{S}} \big\}$.
- **Preference order**: If a virtual node $\ell_{i,j} \in \mathcal{L}$ chooses to compare the performance of any two virtual nodes in $\mathcal{L}_{i,j}$, say $\ell_{s_1,d_1}$ and $\ell_{s_2,d_2}$, its preference order is defined as $\ell_{s_1,d_1} \succ_{i,j} \ell_{s_2,d_2}$ if $\ell_{s_1,d_1}$ is determined to have better performance. Here the performance refers to the key generation rate defined in Eq. (2) for one relay or in Eq. (3) for two relays.

Similar to Nash equilibrium in a non-cooperative game, the *core* plays an essential role in a coalitional game. Generally speaking, the core refers to a set of cooperation strategies such that no coalition can deviate and improve for all its members by cooperation within the coalition [11]. The core of our game $\Omega$ is a set of contributor selection strategies $r_{i,j} \in \Theta(\mathcal{L})$ where there are no coalition $\mathcal{S}$ and $\tilde{r}_{i,j} \in \Theta(\mathcal{S})$ such that $\tilde{r}_{i,j} \succ_{i,j} r_{i,j}$ for all $\ell_{i,j} \in \mathcal{S}$. It means that no improvement on the key generation rate can be made by cooperation within the coalition $\mathcal{S}$. We can prove the existence of a core solution to game $\Omega$. Due to space limitations, we omit the proof here and refer interested readers to [12] for details.

### *C. Core Discovery Algorithm*

This section introduces our core discovery algorithm. For this purpose, we first introduce two concepts as follows.

**Definition 1.** **(Coalitional Subgame)** *Given a coalitional game $\Omega = \langle \mathcal{L}, \mathcal{X}_{\mathcal{L}}, \Theta, (\succ_{i,j})_{\ell_{i,j} \in \mathcal{L}} \rangle$, we call a coalitional game $\Psi = \langle \mathcal{M}, \mathcal{X}_{\mathcal{M}}, \Theta, (\succ_{i,j})_{\ell_{i,j} \in \mathcal{M}} \rangle$ a coalitional subgame of $\Omega$ iff $\mathcal{M} \subseteq \mathcal{L}$ and $\mathcal{M} \neq \emptyset$.*

**Definition 2.** **(Contributor Cycle)** *Given a coalitional subgame $\Psi = \langle \mathcal{M}, \mathcal{X}_{\mathcal{M}}, \Theta, (\succ_{i,j})_{\ell_{i,j} \in \mathcal{M}} \rangle$, a sequence of virtual nodes, $(\ell_{i_1,j_1}, \ldots, \ell_{i_H,j_H})$, is called a contributor cycle of length $H$ if and only if $r_{i_x,j_x} = \ell_{i_{x+1},j_{x+1}}$ for $\forall x \in [1, H-1]$ and $r_{i_H,j_H} = \ell_{r_1,j_1}$.*

A contributor cycle of length one clearly contains a single virtual node, which means that the two mobile nodes forming this virtual node cannot find a relay and thus should directly generate a secret key using noncooperative key generation in Section II-A. In contrast, a contributor cycle of length $H \geq 2$ means that every virtual node in the contributor cycle has its most preferred contributor as the next virtual node of the same cycle in the circular fashion. Every contributor cycle thus corresponds to a coalition, in which the mobile nodes involved are reciprocal for key generation.

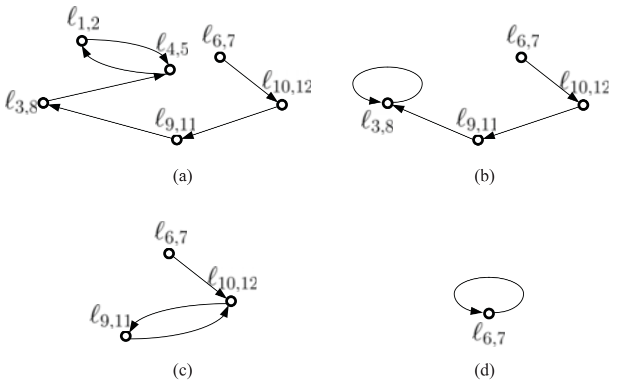
Our core discovery algorithm is to iteratively identify all the contributor cycles which form a core solution. We achieve this by first constructing a directed graph $\mathbf{G} = (\mathcal{L}, \mathcal{E})$, where a directed edge from any vertex $\ell_{i,j}$ to another vertex $\ell_{s,d}$ exists if and only if $\ell_{s,d}$ is the most preferred contributor of $\ell_{i,j}$, i.e., $r_{i,j}^{*} = \ell_{s,d}$. Recall that every virtual node can choose at most one contributor, which corresponds to at most two relays. The outdegree of every vertex $\mathbf{G}$ is thus one if it has at least one candidate contributor and zero otherwise. The problem of discovering all the contributor cycles or the core solution can then be translated into simple-cycle search in $\mathbf{G}$. In particular, a path in a graph refers to a sequence of edges which connect a sequence of vertices, a cycle is a path with the same start and end vertices, and a cycle with no repeated vertices or edges except the start and end vertices is called a simple cycle. Since every vertex in $\mathbf{G}$ corresponds to a virtual node, a simple cycle is equivalently a contributor cycle. So we can denote a simple path of $H$ vertices by a contributor cycle of $H$ virtual nodes as $(\ell_{i_1,j_1}, \ldots, \ell_{i_H,j_H})$. If there is a simple path of $H = |\mathcal{L}| = N/2$ vertices, all the $N/2$ virtual nodes or $N$ mobile nodes are in a single coalition. In contrast, any simple path of one vertex (i.e., a self contributor cycle) means that the two mobile nodes related to the vertex do not use any relay for secret key generation. The following proposition underlies our simple-cycle (contributor-cycle) search algorithm.

**Proposition 1.** *A simple path beginning from any vertex in the directed graph $\mathbf{G}$ results in one and only one simple cycle.*

*Proof:* It is easy to prove that a simple path beginning from any vertex in $\mathbf{G}$ must lead to a simple cycle, as otherwise there must be infinite vertices in $\mathbf{G}$. Now we prove the uniqueness of the resulting simple cycle. If multiple simple cycles exist, there must be at least one vertex whose outdegree

| Virtual Node | Preferred Real Node | Preferred Virtual Node |
|---|---|---|
| $\ell_{1,2}$ | 4 | $\ell_{4,5}$ |
| $\ell_{3,8}$ | 4 | $\ell_{4,5}$ |
| $\ell_{4,5}$ | 2≻9 | $\ell_{1,2} \succ \ell_{9,11}$ |
| $\ell_{9,11}$ | 8≻(12,10) | $\ell_{3,8} \succ \ell_{10,12}$ |
| $\ell_{10,12}$ | (9,11) | $\ell_{9,11}$ |
| $\ell_{6,7}$ | 10 | $\ell_{10,12}$ |

TABLE I: A preference-order table, where $(i, j)$ means both $i$ and $j$ serve as a relay for the corresponding virtual node.



(a) (b)

(c) (d)

Fig. 3: Illustration of contributor cycle discovery.

is larger than one. This contradicts with the property of $\mathbf{G}$ that the outdegree of every vertex is no more than one. ■

Another way to interpret Proposition 1 is that every vertex (virtual node) in $\mathbf{G}$ is on one and only one contributor cycle, possibly a self cycle involving itself only. Then we can discover all the contributor cycles and thus implement the core solution to game $\Omega$ as follows. Initially, all the vertices in $\mathbf{G}$ are marked unvisited. We can start a walk from any unvisited vertex, say $\ell_{i,j}$, until when the walk either hits an visited vertex or returns to $\ell_{i,j}$. In the former case, $\ell_{i,j}$ is marked visited and forms a self contributor cycle. If the later case occurs, a new contributor cycle is found, and all the vertices on the cycle are marked visited. This process continues until either when all the vertices in $\mathbf{G}$ are marked visited or when none of the remaining unvisited vertices can be the start of a walk towards unvisited vertices. In the later case, we mark all the remaining vertices visited and terminate the algorithm.

## V. Implementations

In this section, we present C-SYNERGY and D-SYNERGY, two protocols to implement SYNERGY in centralized and distributed fashions, respectively. We also analyze the security, and quantify computational overhead and communication overhead of C-SYNERGY and D-SYNERGY.

### A. C-SYNERGY: A Centralized Implementation

In C-SYNERGY, every peer pair reports its own preference order to a single server which computes all the contributor cycles and returns each to the corresponding nodes. The server can be a base station if available or a mobile node elected from the mobile nodes themselves. It is worth emphasizing that the server merely does the computation and does not participate in cooperative key generation in the server's role. So it is blind to the final secret key of any peer pair.

*1) Detailed Operations:* Recall that $\mathcal{N} = \{1, \ldots, N\}$ denote the $N$ nodes involved. C-SYNERGY works as follows.

1. Every node $i \in \mathcal{N}$ locally broadcasts a HELLO message including its ID and also records the IDs in received hello messages. Let $\mathcal{C}_i$ denote the neighbor IDs of node $i$.
2. Every two neighboring nodes estimate the channel response between them by exchanging probe messages. The estimated variance is needed for deriving the optimal key rates according to Eq. (2) or Eq. (3).
3. Every two peers, say $i, j \in \mathcal{N}$, exchange $\mathcal{C}_i$ and $\mathcal{C}_j$ to identify their common neighbors as $\mathcal{C}_{i,j} = \mathcal{C}_i \cap \mathcal{C}_j$.
4. Every node $i \in \mathcal{N}$ locally broadcasts its peer ID and also records the peer ID of every neighbor. The peer IDs allow $i$ and its peer $j$ to learn their local topology and the associated peer pairs, based on which to construct the list of candidate contributors, i.e., $\mathcal{L}_{i,j}$.
5. Every two peers, say $i, j \in \mathcal{N}$, compute the optimal key rate for each candidate contributor in $\mathcal{L}_{i,j}$ based on Eq. (4). Then they determine the most preferred contributor which is reported by either of them to the server.
6. The server applies the core discovery algorithm in Section IV-C on the received information to compute all the contributor cycles and finally returns every contributor cycle to each node in that cycle.
7. The nodes in every contributor cycle work together to derive their respective secret key as in Section II-B.

*2) An Example:* We shed more light on C-SYNERGY using the example in Fig. 2, where every two peers $i, j$ are represented with a solid line and annotated by the corresponding virtual node $\ell_{i,j}$. Every two peers jointly determine the preference order for their candidate contributors. We assume that the preference orders are as given in Table I. For instance, $\ell_{1,2}$ has only node 4 as a candidate relay, so its most preferred relay (or contributor) is simply node 4 (or virtual node $\ell_{4,5}$). In addition, $\ell_{9,11}$ has a preference order $8 \succ (12, 10)$, which means that it prefers node 8 as a relay and equivalently virtual node $\ell_{3,8}$ as a contributor.

Based on the received preference orders, the server applies the core discovery algorithm in Section IV-C to derive the contributor cycles. Specifically, the server first constructs a directed graph with six vertices $\{\ell_{1,2}, \ell_{3,8}, \ell_{4,5}, \ell_{9,11}, \ell_{10,12}, \ell_{6,7}\}$. Since $\ell_{4,5}$ is reported as the most preferred contributor of $\ell_{1,2}$, the server adds an edge from $\ell_{1,2}$ to $\ell_{4,5}$ in $\mathbf{G}$. Other edges of $\mathbf{G}$ are added similarly. The resulting graph $\mathbf{G}$ is shown in Fig. 3(a). In iteration 1, the server identifies one contributor cycle $(\ell_{1,2}, \ell_{4,5})$ and removes it from $\mathbf{G}$. The modified graph is shown in Fig. 3(b). In iteration 2, the server identifies a self cycle consisting of $\ell_{3,5}$ only and also removes it. Subsequently, a cycle $(\ell_{9,11}, \ell_{10,12})$ is identified in iteration 3, and a self cycle containing $\ell_{6,7}$ only is identified in iteration 4. Therefore, there are four contributor cycles or coalitions in total, including $(\ell_{1,2}, \ell_{4,5})$, $(\ell_{3,5})$, $(\ell_{9,11}, \ell_{10,12})$, and $(\ell_{6,7})$. Finally, the server returns every contributor cycle to every node involved in that cycle. The mobile nodes can then determine which nodes they can use as a relay and whom they should act as a relay for. For example, nodes 9 and 11 use both nodes 10 and 12 as a relay, and nodes 10 and 12 use both nodes 9 and 11 as a relay. They can all assure that collaborating with each other is the best strategy to improve their respective key generation rate.

### B. D-SYNERGY: A Distributed Implementation

D-SYNERGY enables the mobile nodes to discover the core solution (or contributor cycles) in a purely distributed fashion. To emulate centralized core discovery in C-SYNERGY, D-SYNERGY also works in iterations. Every iteration is initiated by a mobile node not in any identified contributor cycle, and a new contributor cycle is identified in every iteration. D-SYNERGY terminates when every node in $\mathcal{N}$ is included in a contributor cycle.

*1) Detailed Operations:* We introduce two binary flags $f_i$ and $v_i$ for each node $i \in \mathcal{N}$. Referred to as an *inclusion flag*, $f_i$ is initially zero and permanently set to one after a contributor cycle including $i$ is discovered. In contrast, $v_i$ is called a *visit flag* and equals zero if node $i$ has not been included in any contributor cycle at the beginning of an iteration. It is set to and remains one when node $i$ receives a core-discovery message in an iteration. In addition, $v_i$ remains one after node $i$ is included in any contributor cycle. D-SYNERGY works as follows.

1. All the mobile nodes in $\mathcal{N}$ act according to the first five steps of C-SYNERGY.
2. An iteration starts when any node $i \in \mathcal{N}$ with $f_i = 0$ broadcasts a BUSY message. The BUSY message reaches other nodes in $\mathcal{N}$ in a hop-by-hop fashion. Every node resets its visit flag to zero after seeing the BUSY message unless its inclusion flag is one. Multiple nodes may try to initiate an iteration simultaneously, in which case the one with a smaller ID always wins.
3. Node $i$ then sends a REQ$_i$ message to its peer, say node $j$, and also the most preferred relay. Then $i$ and $j$ sets $v_i = 1$ and $v_j = 1$, respectively.
4. Any node $s \neq i$ may receive REQ$_i$ from its peer or a node considering it the most preferred relay. In the former case, node $s$ does nothing other than recording REQ$_i$ and setting $v_s = 1$, as its peer has taken care of REQ$_i$. This operation is designed because $s$ and its peer form a single virtual node in the directed graph $\mathbf{G}$ used in C-SYNERGY. So we let node $s$ and its peer have synchronized internal states to emulate a single virtual node in $\mathbf{G}$. In the latter case, node $s$ does the following operations in sequel.
    - If $f_s = 1$, node $s$ sends a REJ message to the node who sent it REQ$_i$. The receiver of a REJ message then sends REQ$_i$ to its next most preferred relay. If all its candidate relays respond with a REJ message, it returns a REJ message to its own neighbor which sent it REQ$_i$. If $i$ gets REJ from all its candidate relays, it sets $f_i = 1$, notifies its peer $j$ to set $f_j = 1$, and broadcasts an EXIT message to terminate this iteration. In this case, $i$ and $j$ have to establish a secret key without using any relay, which corresponds to the case that the virtual node $\ell_{i,j}$ in $\mathbf{G}$ belongs to a self contributor cycle containing $\ell_{i,j}$ only.
    - If $f_s = 0$ and $v_s = 1$, node $s$ checks whether it has seen REQ$_i$ before. If so, a new contributor cycle is discovered. Node $s$ then notifies every node in this contributor cycle which can be obtained from REQ$_i$. Subsequently, all the nodes in the contributor cycle set their inclusion flag to one. Finally, node $s$ broadcasts an EXIT message to terminate this iteration.
    - If $f_s = 0$ and $v_s = 0$, node $s$ appends $s$ and its peer ID $d$ to REQ$_i$, sets $v_s = 1$, and then sends the modified REQ$_i$ to node $d$ and also most preferred relay.

D-SYNERGY terminates when all the nodes in $\mathcal{N}$ have their inclusion flags set to one.

*2) An Example:* We still use the example in Fig. 2 and the preference orders in Table I to clarify D-SYNERGY.

Assume that node 9 starts the first iteration, in which the inclusion flags $\{f_i\}_{i=1}^{12}$ and visit flags $\{v_i\}_{i=1}^{12}$ are all zero initially. Node 9 sends REQ$_9$ to its peer (node 11) and its most preferred relay (node 8). Then nodes 9 and 11 set $f_9 = 1$ and $f_{11} = 1$, respectively. After receiving REQ$_9$, node 8 finds that $f_8 = 0$ and $v_8 = 0$. So node 8 sends $\langle \text{REQ}_9 \parallel (3,8)\rangle$ to its peer (node 3) and most preferred relay (node 4). Next, nodes 8 and 3 set $v_8 = 1$ and $v_3 = 1$, respectively.

After receiving REQ$_9$ $\parallel (3,8)$, node 4 finds that $f_4 = 0$ and $v_4 = 0$. So node 4 sends $\langle \text{REQ}_9 \parallel (3,8) \parallel (4,5)\rangle$ to its peer (node 5) and most preferred relay (node 2). Next, nodes 4 and 5 set $v_4 = 1$ and $v_5 = 1$, respectively. Similarly, node 2 sends $\langle \text{REQ}_9 \parallel (3,8) \parallel (4,5) \parallel (1,2)\rangle$ to its peer (node 1) and most preferred relay (node 4). Also, nodes 2 and 1 set $v_2 = 1$ and $v_1 = 1$, respectively.

After receiving $\langle \text{REQ}_9 \parallel (3,8) \parallel (4,5) \parallel (1,2)\rangle$, node 4 finds that $f_4 = 0$ and $v_4 = 1$. In addition, it has seen REQ$_9$ before, so there is a contributor cycle including peer pairs $(4,5)$ and $(1,2)$, which can also be represented by virtual nodes $(\ell_{1,2}, \ell_{4,5})$. Then node 4 broadcasts the contributor cycle and an EXIT message to all the other nodes. Subsequently, the nodes 1, 2, 4, and 5 all have their inclusion flag set to one. Finally, all the remaining nodes, i.e., $\{3,8,9,11,6,7,10,12\}$, set their visit flag to zero and enter the next iteration. This process continues until finding three other contributor cycles as $(\ell_{3,5})$, $(\ell_{9,11}, \ell_{10,12})$, and $(\ell_{6,7})$.

### C. Performance Analysis

In this section, we analyze the security, computational overhead, and communication overhead of SYNERGY (C-SYNERGY and D-SYNERGY).

**Security Analysis:** The security of the generated secret key is first guaranteed by the key generation process introduced in Section II-B. The generated secret key is provably secure from any eavesdropper who experiences an independent wireless channel from the legitimate nodes [6]. In addition, neither C-SYNERGY nor D-SYNERGY discloses any secret-key information to eavesdroppers. Although eavesdroppers might overhear the candidate relay nodes and the preference order of each peer pair who want to establish a secret key, they cannot be in the proximity of any legitimate node and thus still cannot extract any useful information from the wireless channel. Furthermore, although a relay node knows partial information about a secret key, it is blind to the rest information tied to the wireless channel between the peer nodes it assists. As note that SYNERGY can be easily adapted to work with

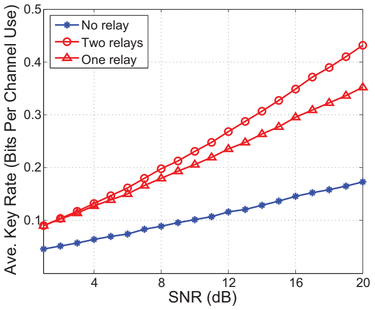


Fig. 4: Average key rate for 20 users.

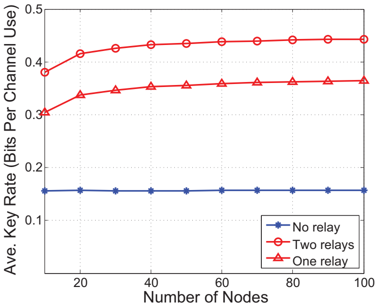


Fig. 5: Average key rate for $D = 200$ m.

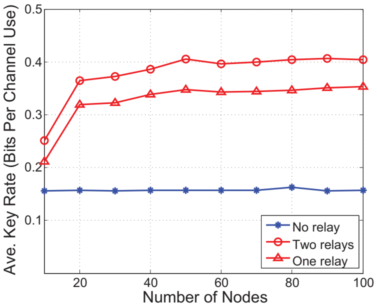


Fig. 6: Average key rate for $D = 300$ m.

the most advanced cooperative key generation technique in [6] such that the relay nodes know nothing about the final secret key. We finally want to point out that C-SYNERGY and D-SYNERGY are both vulnerable to active attacks on modifying the information exchange to and from every mobile node. Such active attacks can be mitigated, e.g., by authenticating the exchanged information using a temporal group key chosen by any involved node. Same as all previous work on PHY-based secret key generation, we focus on passive eavesdropping attacks in this paper. A detailed treatment of active attacks is beyond the scope of this paper.

**Computational Overhead:** SYNERGY's computational overhead is mainly incurred in the process of discovering the contributor cycles or the core solution. In particular, according to the description of the core discovery algorithm in Section IV-C, we can easily see that the computation complexity of SYNERGY is $\mathcal{O}(|\mathcal{L}_j|)$ at $j$th iteration, where $|\mathcal{L}_j|$ denotes the number of virtual nodes involved in the $j$th iteration. Therefore, the overall computation complexity for all $J$ iterations is $\mathcal{O}(\sum_{j=1}^{J} |\mathcal{L}_j|)$. Although $|\mathcal{L}_j|$ and $J$ depend on many factors and cannot be precisely determined, we can estimate the lower and upper bounds for the computational complexity. Specifically, the lower bound is $\mathcal{O}(N)$ which is achieved when all the $N/2$ virtual nodes form a single contributor cycle in the first iteration. In contrast, the upper bound is attained if every virtual node is fond to form a self contributor cycle, leading to $N/2$ iterations in total. This corresponds to an upper bound $\mathcal{O}(\sum_{j=1}^{J} |\mathcal{L}_j|) = \mathcal{O}(\sum_{i=1}^{N/2} i) = \mathcal{O}(N^2)$. The computations are performed at a single server in C-SYNERGY and distributed over mobile nodes in D-SYNERGY.

**Communication Overhead:** The communication overhead of two SYNERGY implementations is different. Specifically, the communication overhead of C-SYNERGY is mainly incurred in channel estimation, neighbor discovery, and communication between mobile nodes and the server. It can be estimated by $\mathcal{O}(\tilde{N}^2)$, where $\tilde{N}$ is the average number of neighbors every node has. In addition to the above overhead, D-SYNERGY incurs some message overhead in the distributed core-discovery phase. Its communication overhead can be lower-bounded by $\mathcal{O}(N)$, which occurs when all the virtual nodes form a contributor cycle in one iteration, and upper-bounded by $\mathcal{O}(N^2)$, which is incurred when each iteration produces a self cycle containing a unique virtual node. Therefore, the overall communication overhead of D-SYNERGY is larger than that of C-SYNERGY. So we can prefer C-SYNERGY to D-SYNERGY unless a base station does not exist, and no mobile node can be elected as a server.

## VI. Performance Evaluation

In this section, we evaluate the performance of C-SYNERGY and D-SYNERGY via Matlab simulations. The simulation strategy and settings are as follows. We consider a square region with a side length of $D$ meters. We randomly deploy $N$ nodes in the square region and assume that the transmission range of each node is a circle of radius $A$ meters. We set coherent time $T = 20$ symbols, and the Gaussian noise variance $\sigma^2 = 1$. Then the channel variances between every two neighboring nodes are set to be a random variable with uniform distribution. In addition, we randomly select two nodes within each other's transmission range as a peer pair to establish a secret key. Each point in the following figures represents the average value of 1000 runs. Since the results for C-SYNERGY and D-SYNERGY are the same in Figs. 4∼7, we do not differentiate them there.

Fig. 4 illustrates the impact of SNR on the optimal key rates with no relay, with one relay, and with two relays. The first case corresponds to non-cooperative key generation, and the later two cases correspond to SYNERGY (cooperative key generation). In this set of simulations, we fix the region side length $D = 200$ m and set the transmission range of each node large enough to cover the whole square region. Besides, we fix the number of users to $N = 20$ and increase SNR from 1 to 20 dB. From Fig. 4, we can clearly see that as SNR increases, the optimal key rates of the three cases all increase. This is anticipated because the key generation rate increases with the transmission power according to Eqs. (1)∼(3). Moreover, the optimal key rate of SYNERGY always outperforms non-cooperative key generation, and it is always better to use two relays (if any) than using one relay. This is also as expected because the more relays, the more common channel randomness available for secret key generation.

Fig. 5 demonstrates the impact of the average number of nodes on the optimal key rate. In this set of simulations, we fix SNR = 20 dB, the transmission range of each node to 200 m, and the region side length to $D = 200$ m. We also vary the number of nodes from $N = 10$ to 100. We can observe that the optimal key rate of SYNERGY is much higher than that of non-cooperative key generation. In addition, the optimal key rate of non-cooperative key generation is almost stable along with the increase of users, as it only depends on the channel condition between two peer nodes who want to generate a secret key and does not rely on any other node. In contrast, the more nodes in a fixed region, the more candidate relay

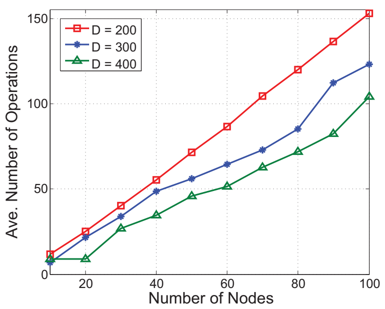


Fig. 7: Comp. overhead (SYNERGY).

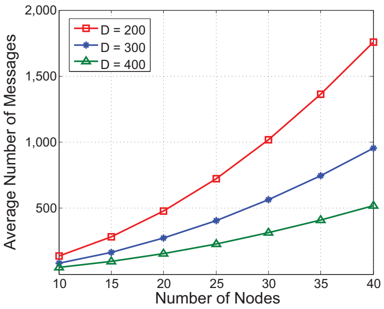


Fig. 8: Comm. overhead (C-SYNERGY).

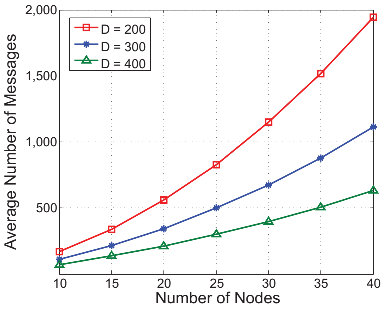


Fig. 9: Comm. overhead (D-SYNERGY).

nodes available for two peer nodes. So we can observe that the optimal key rate of SYNERGY increases with the number of users.

Fig. 6 shows the impact of the average number of neighbors on the optimal key rate when the region side length is $D = 300$ m. Other simulation settings are the same for generating Fig. 5, so the entire region is larger than every node's transmission range. We have almost the same observations as in Fig. 5 due to the same reason. In addition, the optimal key rate of SYNERGY in Fig. 6 is always lower than that in Fig. 5 for the same $N$, as the larger the region, the less likely that two peer nodes can find a common neighbor as a relay node. Another observation is that the gap between the optimal key rates of the one-relay and two-relay cases becomes smaller in contrast to Fig. 5. The reason is that a larger region makes it more difficult for two peer nodes to find two common neighbors as two relay nodes who themselves need to be two peer nodes as well according to our requirement in SYNERGY. Moreover, when there are fewer than 20 users, the optimal key rate of SYNERGY is only slightly better than non-cooperative key generation, as most peer pairs cannot find a relay in their common transmission range. Finally, it is still better to use two relays than using one relay in SYNERGY.

Fig. 7 shows the impact of the number of nodes on the average number of operations for discovering all contributor cycles. In this set of simulations, we fix SNR = 20 dB and each node's transmission range to 200 m. As we can see, the average number of operations needed increases almost linearly with number of nodes. Since the overall computational overhead of SYNERGY is dominated by contributor-cycle discovery, this result confirms the high computational efficiency of SYNERGY. As said, the computational overhead of SYNERGY is incurred at a single server in C-SYNERGY but distributed over the $N$ mobile nodes in D-SYNERGY. In addition, we compare the average number of operations needed when the region side length $D = 200$ m, 300 m, and 400 m. For a fixed number of nodes, the larger the region, the fewer common neighbors and thus candidate relay nodes every two peer nodes have, the fewer edges in the graph $\mathbf{G}$ composed of virtual nodes, and the fewer operations needed for contributor-cycle discovery. This conjecture is confirmed in Fig. 7.

Fig. 8 demonstrates the impact of the number of nodes on the communication overhead of C-SYNERGY. In this set of simulation, we fix SNR = 20 dB and each node's transmission range to 200 m. The communication overhead lies in the messages for neighbor discovery, channel estimation, and communicating with the server. The total number of messages and thus the communication overhead obviously would increase with the number of nodes, as shown in Fig. 8. For a given number of nodes, the larger the region, the fewer neighbors each node has, and the fewer messages needed for neighbor discovery and channel estimation. So we can see that the communication overhead of C-SYNERGY decreases as the region side length $D$ increases.

Fig. 9 illustrates the impact of the number of nodes on the communication overhead of D-SYNERGY under the same simulation settings for Fig. 8. The simulation results show the similar trend in Fig. 8 due to the similar reason. One point we want to point out is that a larger region for a given number of nodes can decrease the likelihood that two peer nodes find a relay node in their common communication range, leading to possibly fewer edges between virtual nodes in the directed graph $\mathbf{G}$. As such, the number of messages incurred in channel estimation and distributed relay-cycle discovery is likely to be reduced. This factor also contributes to the reduced communication overhead in Fig. 9 as the region side length $D$ increases. Furthermore, D-SYNERGY has higher communication overhead than C-SYNERGY due to distributed contributor-cycle discovery, but it does not need a base station or an elected node as a server doing centralized computation.

As a summary of the above simulation results, the more neighbors each node has, the higher the optimal key generation rate, and the higher the computational and communication overhead of SYNERGY. There is thus an inherent tradeoff between the key generation rate and the associated computational/communication overhead.

## VII. Related Work

In this section, we briefly discuss some work most germane to SYNERGY, which is divided into two categories.

**Secret Key Generation from Wireless Channels.** There has been tremendous effort on exploring the channel reciprocity to establish a secret key between two mobile nodes. For example, the work in [4] focuses on using spatial and temporal variations of the wireless channel, while [5] focuses on exploring multi-antenna diversity for secret bit extraction. The work in [13] aims at group key establishment in star and chain networks, and the work in [14] targets secret key establishment in body area networks. The channel information used in [4], [5], [13], [14] is RSS (Received Signal Strength). In contrast, the work in [3] tries to extract a secret key from the channel response between two wireless devices. There is also research on using the phase change of received signals for secret key generation in UWB systems [15] and OFDM systems [16], respectively. This line of work [3]–[5], [13]–[16] can be regarded as different realizations of the information-theoretical approach in [9] and lead to different approximations to the optimal key rate in Eq. (1). In addition, this line of work [3]–[5], [13]–[16] belongs to non-cooperative key generation, as only the direct wireless channel between two wireless devices is explored. As such, the key generation rate in [3]–[5], [13]–[16] can be very low in slowly changing wireless environments. In contrast, SYNERGY has a different focus on stimulating mobile nodes in helping others establish a secret key to get help in return. Once SYNERGY identifies the contributor cycles, the techniques in [3]–[5], [13]–[16] can all be adopted to establish a secret key between two mobile nodes as well as between each node and each of their relays. The resulting keys can finally be combined to produce the actual secret key between the two nodes, as illustrated in Section II-B.

Cooperative key generation [6], [8], [17] is a relatively new research topic. The work in [6], [8] investigates relay-assisted strategies to improve the key generation rate by incorporating additional randomness brought by the relay nodes who are the common neighbors of two mobile nodes under consideration. In addition, the work in [17] studies secret key generation in a two-way relay channel, where there is no direct wireless channel between two mobile nodes who want to establish a secret key. A critical issue that has been overlooked in [6], [8], [17] is that mobile nodes are selfish in nature and will not act as relays for others without adequate reward in return. SYNERGY fills this great void.

**Cooperative Communication via Social Reciprocity.** SYNERGY is motivated by the recent work on cooperative communication [18]. Specifically, the work in [18] targets a multi-hop D2D communication scenario, in which each node can choose to serve as a relay for other nodes. A novel coalitional game-theoretical framework is developed to design cooperation strategies based on social trust and social reciprocity. The authors prove the existence of a core solution and propose a mechanism to implement the core solution by identifying reciprocal cycles, each of which contains the nodes motivated to act as relays for others in the same cycle. In contrast to [18], SYNERGY focuses on cooperative key generation, a very different problem. In addition, the game formulation in [18] cannot be directly applied, as each game player in our scenario corresponds to two nodes instead of one as in [18]. Moreover, each node in [18] can use at most one relay node in its vicinity, while each node in SYNERGY can use two relays to achieve a higher key generation rate than using one relay.

## VIII. Conclusion

In this paper, we studied secret key establishment, a fundamental challenge for securing D2D communication. We proposed SYNERGY, a game-theoretical approach for stimulating PHY-based cooperative key generation in wireless networks as the first work of its kind. In SYNERGY, incentive-aware cooperative key generation is formulated as a coalitional game. We designed centralized and distributed protocols for finding a core solution to the coalitional game. With SYNERGY in place, selfish mobile nodes are strongly motivated to collaborate with others in the same coalition to improve their respective key generation rate. The efficacy and efficiency of SYNERGY has been confirmed by extensive simulations.

## Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-1117462, CNS-0844972 (CAREER) and CNS-1320906. We would also like to thank anonymous reviewers for their constructive comments and helpful advice.

## References

[1] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 96–104, June 2012.

[2] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *CCS'07*, Alexandria, Virginia, USA, 2007.

[3] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MobiCom'08*, San Francisco, California, USA, 2008, pp. 128–139.

[4] S. Jana, S. Premnath, M. Clark, S. Kasera, N. Patwari, and S. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *MobiCom'09*, Beijing, China, 2009.

[5] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, "Exploiting multiple-antenna diversity for shared secret key generation in wireless networks," in *Proceedings of the 29th conference on Information communications*, ser. INFOCOM'10, San Diego, California, USA, 2010.

[6] L. Lai, Y. Liang, and W. Du, "Cooperative key generation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 8, pp. 1578–1588, 2012.

[7] S. Gollakota and D. Katabi, "Physical layer wireless security made fast and channel independent," in *INFOCOM'11*, Shanghai, China, Apr. 2011.

[8] L. Lai, Y. Liang, and W. Du, "Phy-based cooperative key generation in wireless networks," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, Allerton House, UIUC, Illinois, USA, 2011.

[9] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography. i. secret sharing," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1121–1132, 1993.

[10] T. Cover and J. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.

[11] R. Myerson, *Game theory: analysis of conflict*. Harward University Press, 1997.

[12] J. Sun, X. Chen, Y. Zhang, and J. Zhang, "SYNERGY: A game-theoretical approach for cooperative key generation in wireless networks," Arizona State University, Technical Report, July 2013, http://cnsg.asu.edu/files/sun-INFOCOM14.pdf.

[13] H. Liu, J. Yang, Y. Wang, and Y. Chen, "Collaborative secret key extraction leveraging received signal strength in mobile wireless networks," in *IEEE INFOCOM'12*, Orlando, FL, Mar. 2012.

[14] S. Lu, J. Yuan, S. Yu, and M. Li, "ASK-BAN: authenticated secret key extraction utilizing channel characteristics for body area networks," in *ACM WiSec'13*, Budapest, Hungary, Apr. 2013.

[15] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Trans. on Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, Sep. 2007.

[16] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *ICASSP'08*, 2008.

[17] H. Zhou, L. Huie, and L. Lai, "Key generation in two-way relay wireless channels," in *CISS'13*, Baltimore, MD, USA, 2013.

[18] X. Chen, B. Proulx, X. Gong, and J. Zhang, "Social trust and social reciprocity based cooperative d2d communications," in *MobiHoc'13*, Bangalore, India, 2013.