Securing Recommendations in Grouped P2P E-Commerce Trust Model

Felix Musau, Guojun Wang*, *Member, IEEE,* Shui Yu, *Member, IEEE,* and Muhammad Bashir Abdullahi

***Abstract*—In dynamic peer to peer (P2P) e-commerce, it is an important and difficult problem to promote online businesses without sacrificing the desired trust to secure transactions. In this paper, we address malicious threats in order to guarantee secrecy and integrity of recommendations exchanged among peers in P2P e-commerce. In addition to trust, secret keys are required to be established between each peer and its neighbors. Further, we propose a key management approach *gkeying* to generate six types of keys. Our work mainly focuses on key generation for securing recommendations, and ensuring the integrity of recommendations. The proposed approach presented with a security and performance analysis, is more secure and more efficient in terms of communication cost, computation cost, storage cost, and feasibility.**

***Index Terms*—Peer to peer (P2P); key management; key generation; trust; security.**

I. Introduction

P2P e-commerce is a computing application in which a peer communicates directly with others to exchange information, to inquire on products and services, or to execute business transactions. This poses a potential threat from malicious peers, as a peer rarely has any prior information about others. In P2P e-commerce, key management has been used for the safety, storage, and transmission of information in the presence of malicious peers and other threats. P2P and other decentralized, distributed systems are known to be particularly vulnerable to sybil attacks [1]. Indeed, two types of keys have been used in many applications, i.e., encryption and decryption keys. Key pre-distribution has a drawback in which an attacker may know the distribution of the polynomial shares on transit. As a result, an attacker may precisely target certain peers, in an attempt to learn the shares of a particular bivariate polynomial. In existing models, key management has been addressed among peers in general networks, with little emphasis on continuous exchange of trust feedback and recommendations. The damage caused by insecure exchange of recommendations in P2P e-commerce is more than the other applications.

Previous approaches have not been able to address effectively the threats caused by malicious peers as entities and links increase. There is a necessity to ensure security and trust in exchange of recommendation information. To address the threats and risks, we propose a method of generating six types of keys, referred to as *gkeying*. For the integrity of recommendations, Reed Solomon Code is used as an erasure code, and it helps in reliability to recover recommendations that pass through unreliable channels. The peers have to manage threats and risks involved in their transactions with no prior experience and knowledge about each other's reputation. Trust has to be established, hence recommendations exchanged, transmitted, stored, and maintained among the peers in a group, safeguarded by the generated six types of keys. In the proposed approach, trust relationships between peers are divided into four categories: 1) Trust relationship between two peers in the same group; 2) Trust relationships within a peer group; 3) Trust relationships between different groups; and 4) Trust relationships between a peer in a group with another peer outside the group.

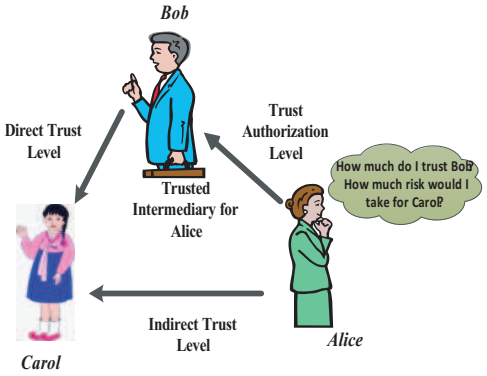


Fig. 1: A direct recommendation-based trust scenario.

A recommendation in our case refers to the feedback and trust evaluation level exchanged among peers. This is generated as trust recommendations based upon each peer's opinion. For example, if Carol wants to trade with Alice for the first time, and they have no idea of each other, then Carol could ask Bob about Alice as shown in Fig. 1. In this case, the information given by Bob about Alice has to be safeguarded against attacks. A minimal change can convince Carol to make wrong decisions assuming it is the true information from Bob. When evaluating the trustworthiness of a given party, a peer combines its local evidence based on direct prior interactions and the testimonies of others known as recommendations. We focus on peers existing in groups based on interest in addition to *gkeying*, to protect the integrity of the recommendations

F. Musau, G. Wang, and M. B. Abdullahi are with the School of Information Science and Engineering, Central South University, Changsha, Hunan Province, P. R. China, 410083.

S. Yu is with the School of Information Technology, Deakin University, 221 Burwood Highway, Burwood Victoria 3125, Australia.
F. Musau is also with the School of Engineering and Technology, Kenyatta University, 43844 Nairobi, Kenya.
*Correspondence to: csgjwang@mail.csu.edu.cn.
Associate editor, Gregorio Martinez Perez.
Manuscript received 26 March, 2012; revised 21 Aug, 2012;

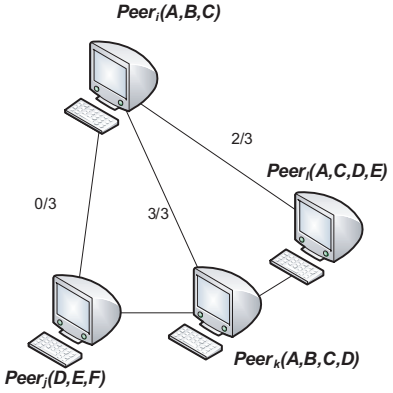


Fig. 2: Interest-based trusted neighbors.

exchanged from malicious peers. The primary goal of our proposed approach is to establish a trusted, confidential, and secure channel among group members although there are differences in trust levels.

The basic step to secure trust information is to provide cryptographic keys. Cryptography has two dominant flavors, namely, symmetric-key (secret-key) and asymmetric-key (public-key) approaches. In the symmetric-key cryptography, the same key is used to encrypt and decrypt recommendations, while in the asymmetric-key approach, different keys are used to convert and recover information [2]. There is a variety of symmetric or asymmetric algorithms available, such as DES, AES, IDEA, RSA, and EIGamal [3], [4]. Threshold cryptography [5] can also be used, a good example is Shamir's $(k; n)$ secret sharing scheme, where a secret is split into n pieces according to a random polynomial.

According to our approach, a peer joining a group ensures that the group peers have same interest, and can be trusted to a certain level. To join a group, a peer has to pass through an admission control process as outlined in the preliminaries. The network environment is therefore operated and managed by the peers, which makes P2P e-commerce depending upon a cooperative and trusting nature [6], [7]. A non-cooperative peer, also called a " free rider ", leaves the network at once when it makes a selfish transaction. We note that a free rider in our context is an abstract description of a non-cooperative behavior, which is different from that in BitTorrent systems. In our approach, we prefer symmetric key techniques in decentralized P2P e-commerce rather than asymmetric key techniques, which require significantly more computation to protect information from malicious entities.

In our approach, to secure all the communications, a key is shared by a peer with each trusted neighbor. In P2P e-commerce, trust relation is used for neighborhood formation [8]. A secret key delivery technique using a multi-hop trusted path is used. A neighbor peer finds a multi-hop secure path toward another. An adversary can easily listen to the traffic, impersonate one of the peers, or intentionally provide fake or misleading feedback. The peers' trust relationships are based on the same interest as shown in Fig. 2. If two peers have no common interest, they have no relation to each other in our proposed approach. In Fig. 2, we consider a trust neighbor relation between *peer*$_i$ and *peer*$_k$, as *peer*$_i$ looks for products of interest. According to Fig. 2, *peer*$_i$ has no common interest with *peer*$_j$. As peers change interest from time to time, therefore, the relationship and neighborhood links change accordingly. The changes necessitate the update of keys.

In our approach, most keys are issued dynamically, and change over time. Our approach improves the network resilience compared to existing schemes. In this paper, we advance the area of P2P networks as it addresses malicious attacks and other threats, which characterizes the behaviors of compromised peers due to their chacteristics of open and anonymous nature. This paper also advances integrity, confidentiality, availability, reliability, and access control policies in decentralized P2P applications. We consider a way to establish a secure routing structure over which recommendations can be exchanged reliably in the presence of malicious peers. Different keys have different roles in the proposed grouped P2P e-commerce structure. The recommendations exchanged are secure as any compromise of any key would lead to key revocation and update of all keys.

Our goal in this paper is to address the above mentioned threats whenever peers exchange recommendations and trust information among each other. Further, we provide a mechanism to identifify and isolate malicious peers.

Our contribution is threefold:

1) The proposed approach *gkeying* is composed of six types of keys. It provides efficient key generation among peers in exchange of recommendations in P2P e-commerce environment. This approach can be used to address threats from malicious peers.
2) The proposed approach can accommodate dynamic groups of peers efficiently, while preserving anonymity. It maintains the properties of forward and backward secrecy among group members. In addition, it can resist against threats and risks in the presence of malicious peers.
3) Recommendations exchanged can be recovered in case of any errors depending on a certain threshold. We improve network resilience against peer capture and other attacks.

Organization of the paper. First, we introduce related work in Section II. Next, we give system models in Section III. Section IV shows preliminaries. The proposed approach is presented in Section V. Section VI deals with attacks. Security and performance analysis is given in Section VII. Section VIII summarizes our conclusions and discusses future work.

II. Related Work

Key management is a fundamental task needed to safeguard and preserve the recommendations exchanged among peers. It involves: 1) How to reduce the overhead of key generation; 2) How to minimize the number of encryptions and decryptions; 3) How to reduce the number of key messages; and 4) How to share a secure group key. Interms of keying relationship peers share keying materials for use in cryptographic mechanisms [9], [10]. The keying materials include public

and private key pairs, secret keys, initialization parameters, and non-secret parameters supporting key management. The fundamental function of key management is the establishment and distribution of keying materials. Key establishment is subdivided into key agreement and key transport. Traditional key management schemes use a key graph to manage all the keys in a group. This enables one key to be shared by many users resulting in the "one-affect-many" problem [12].

For an open message, the only claim the sending side can make is that recommendations were modified on the way to the destination. To prove (or disprove) such a claim, one-way hash of the recommendations is encrypted with a public key before being sent. There are many studies in network security addressing key management, authentication, and vulnerability analysis, which form the basis of our work.

Kim et al. [13] opened investigation on admission control in peer groups and developed a framework using public key infrastructure (PKI) certificates, digital signature, and secret sharing. The work represents an initial attempt to construct an admission control framework suitable for different flavors of peer groups, and match them with appropriate cryptographic techniques and protocols. Their work did not deal with the integration of admission control and group key management.

Sun et al. [14] proposed a multi-group key management scheme (MGKMS). However, the method promotes centralized mechanisms which can not work in decentralized P2P e-commerce. Zhang et al. [8] proposed the evaluation similarity degree under different context of services and gave local and global reputation computation. Their work did not deal with issue of securing the recommendations

Eschenauer and Gligor [15] introduced a key management scheme based on probabilistic key sharing for distributed sensor networks with central key servers. They proposed the use of redeployed keys for encrypting all communications between peers. In their work, a session key between two peers can also be established using a logical path secured by the redeployed keys.

Xiong et al. [16] proposed PeerTrust, which is a dynamic trust model for quantifying and assessing the trustworthiness of peers in P2P e-commerce community. They showed that the interference of recommendations takes place in storage or during transmission, thus they used two layers, PKI based scheme and data replication, to increase security and reliability of recommendation management. For feedback submission, peer v submits the feedback about peer u, signed with its private key $SK(v)$ along with its public key $PK(v)$. Each piece of feedback is signed with the feedback source private key, which guarantees the integrity of the feedback and the authenticity of the feedback origin. Josang et al. [17] proposed trust and reputation systems represent a significant trend in decision support for Internet mediated service provision. The trust provides an incentive for good behavior and therefore tends to have a positive effect on market quality. In the work they did not to address major threats in business transactions.

Zhou et al. [9] proposed a distributed key management system, by using threshold cryptography to distribute trust among a set of servers. The set of servers act as certificate signing authority to sign certificates. The whole network system has a pair of keys in which a public key is distributed to the system. The private key k is divided into n shares using the $(n,\, k+1)$ threshold cryptography scheme. The shares are distributed to n arbitrarily chosen nodes (servers). In order to obtain a certificate, a node contacts $k+1$ servers and each server generates a signature with its share of the private key. The $k+1$ partial signatures are submitted to a combiner to compute the certificate signature. Shares are distributed to a given number of nodes that may be compromised.

Chan et al. [11] proposed a q-composite random key pre-distribution scheme [18] for key management. The difference between their scheme and the earlier one is that q common keys $(q \geq 1)$, instead of just a single one, are needed for secure communications between a pair of nodes. Du et al. [19] proposed a key pre-distribution scheme, which can improve simultaneously the resilience of the network compared to other schemes. This approach has a disadvantage in P2P systems as the use of key pre-distribution is not possible in dynamic, open, and anonymous systems.

Klaoudatou et al. [22] grouped the WSNs application environments into two major categories (infrastructure-based and infrastructureless) and have examined: a) Which of the cluster-based Group Key Agreement(GKA) protocols that appear in the literature are applicable to each category, and b) To which degree the protocols will impact the systems performance.

Wu et al. [23] introduced an asymmetric group key agreement (ASGKA), allowing a set of users to establish a common public encryption key. Their scheme achieves the advantages of both group key agreement and broadcast encryption. However, as the authors claimed, the scheme works only for static groups, and does not consider a member joining or leaving the group. Once there is a member leaving, the system should be reset, including refreshing all users of both public and private keys, hence the communications among the group and its subgroups is broken. The scheme is inefficient if applied in P2P that has rapidly changing membership.

The characteristics of P2P networks have contributed to risk and threats not addressed by existing technologies in distributed systems. Our work combines group key management and admission control that previous schemes failed to address. Most of the earlier schemes were in centralized key management, where they relied more on trusted third parties (TTPs). In our work, every peer participates in generating the keys. The system addresses the change of keys dynamically that previously relied on key graphs. Another key issue our method has been a success is to rely on the cooperation of individual peers other than a key distribution center (KDC) to generate keys. This eliminates the occurrent of malicious peers launching an attack at a central point, which has more impact. A failure of the central point is a failure of the whole system.

In P2P, a symmetric key encryption system enables an encrypted recommendation file to be decrypted by a user using his private key. Shamir introduced the concept of identity-based cryptosystem (IBC) to simplify the certificate management process. In an ID-based setting, the public key is generated by TTP, called a private key generator (PKG). An ID-based system enables peers to communicate without

exchanging private or public keys, and without accessing public directory. Sahai et al. [24] introduced the notion of an attribute-based encryption (ABE) as an extension of ID-based encryption [25], in which a sender encrypts a message, specifying an attribute set and a number *k*, so that only the recipients who have at least *k* attributes of the given attributes can decrypt the message.

When the membership changes, the attribute keys of the changing user should be refreshed. Since one attribute is shared by multiple users, the scheme cannot avoid the "one-affect-many" problem. Goyal et al. [26] extended the idea of ABE and presented two variants: key policy attribute-based encryption (KP-ABE) and ciphertext policy attribute-based encryption (CP-ABE). The first CP-ABE scheme was proposed by Bethencourt et al. [27]. In a KP-ABE system, the private key of a party is associated with an access policy defined over a set of attributes while the ciphertext is associated with a set of attributes.

III. System Models

A. Network Model

We consider a densely populated P2P e-commerce network consisting of N peers. This can be formulated as an undirected graph, $G = (V, E)$, where Vrepresents the set of peers in the network, i.e., $V = \{v_1, v_2, ..., v_n\}$, and E represents TCP or UDP connections between end hosts, i.e., $E = \{e_1, e_2, ...e_m\}$. An edge $e = (u, v)$ exists in G if peer u is allowed to make transactions with peer v via a connection. Each peer maintains a list of unique identifiers of the peers in its neighborhood $N(u) = \{v|(u, v) \in E\}$. Messages can be sent from a peer u to any peer v. P2P e-commerce is assumed to be an overlay on top of an existing physical network (e.g., the Internet). Similar to P2P e-commerce topology models in the literature [28], a peer can establish an overlay connection with any other peers so that G forms a full mesh.

We assume that a peer v_a forms similarity groups based on their business interests. Each peer in a group maintains a peer reputation table that contains the reputation information of all the peers in its neighborhood with respect to direct business transacted. The peer reputation table is updated whenever there is a new observation either directly or indirectly. The reputation information reflects the view or opinion of a peer at a certain points in time about each of its business neighbors.

B. Attack Model

P2P e-commerce communities are day by day gaining acceptance and popularity in the Internet, because they provide an infrastructure where expected products can be located and traded. However, peer characteristics open the door for possible threats, misuses, and abuses by malicious peers. In this paper, we consider a situation where one or more peers may be compromised and act maliciously in the network. The compromised peer(s) may launch the following attacks:

- *Passive attack*: listen to incoming and outgoing messages, in order to infer the relevant information from the transmitted recommendations, i.e., eavesdroppng, but doesn't harm the system. A peer can be in passive mode and later in active mode.
- *Active attack*: when a malicious peer received a recommendation for forwarding, it can modify, or when requested to provide recommendations on another peer, it can inflate or bad mouth. The bad mouthing is a situation where a malicious peer may collude with other malicious peers to launch attacks to honest peers.

IV. Preliminaries

In this section, we give some foundations to form the basis of our approach.

Connected Graph: Let *G* be a connected graph whose vertex set and edge set are *V* (*G*) and *E*(*G*) respectively. The distance between two vertices u and v, denoted by $d_G(u, v)$, is the length of a shortest path between them. Note $|V(G)| = n$ and $|E(G)| = m$, for $u \in V(G), \Gamma(u)$ denotes the set of its neighbors in *G*, and the degree of u is $d_u = |\Gamma(u)|$. If *G* is a connected graph on n vertices, the edge connectivity of G is equal to k in all subgraphs of G, obtained by deleting k edges from G. If k is the edge connectivity of G, then $1 \leq k \leq n-1$ with $k = n-1$, if and only if $G = K_n$. Let G and H be two graphs with *V*(*G*)∩*V*(*H*) = ∅. By *G*∪*H*, we denote the disjoint union of G and H. The join of G and H, denoted by $G+H$, is the graph with vertex set *V*(*G* + *H*) = *V*(*G*)∪*V*(*H*), and edge set $E(G+H) = E(G) \cup E(H) \cup \{uv|u \in V(G), v \in V(H)\}$.

Key Path: A key path between peer A and peer B is defined as a sequence of peers: $(A, N_1), (N_1, N_2), ..., (N_{j-1}, N_j), (N_j, B)$, such that each pair of peers has at least one shared key after the key discovery phase. The length of the key path is the number of pairs of peers in it. Each key is different from the other.

Typically, $\mathbb{G}_1$ is an elliptic-curve group and $\mathbb{G}_2$ is a finite field. Alternatively, let e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map, the Lagrange's polynomial coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set *S* of elements in $\mathbb{Z}_p$ is defined as $\Delta_{i,S} = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. To setup for the public key and master key: Groups $\mathbb{G}_1, \mathbb{G}_2$ are chosen and bilinear defined as e: $\mathbb{G}_1 \times \mathbb{G}_1 = \mathbb{G}_2$. It also selects $\alpha, \beta_1, \beta_2 \in \mathbb{Z}_p$ such that $\beta_1 \neq \beta_2, \beta_1 \neq 0$ and $\beta_2 \neq 0$. The public key is $PK = (\mathbb{G}_0, \mathbb{G}_1, e, g, h_1 = g^{\beta_1}, f_1 = g^{1/\beta_1}, h_2 = g^{\beta_2}, f_2 = g^{1/\beta_2}, e(g, g)^{\alpha})$, and the master key is $(\beta_1, \beta_2, g^{\alpha})$. The key generation algorithm takes as input the master key *MK* and a set of feedback *S*, and outputs a private key corresponding to *S*. It chooses $r, r_{vk} \in \mathbb{Z}_p$, $r_j \in \mathbb{Z}_p$, for each $j \in S$. A private key is computed as: $SK = (D = g^{(\alpha+r)/\beta_1}, E = g^{r/\beta_2}, \forall j \in S : D_j = g^r . H(j)^{rj}, D'_j = g^{rj})$. A recursive algorithm *DecryptPeer* (*C, SK, x*) that takes as input *C*, as a private key *SK* associated with a set of attributes *S* and a node x from a group. If x is a peer, then let $i = att(x)$. If $i \notin S$, then let $DecryptPeer(C, SK, x) = \perp$, and it is then defined as: $DecryptPeer(C, SK, x) = \frac{e(Di,Cx)}{e(D'_i,C'_x)} = \frac{e(g^r . H(i)^{r_i} . g^{q_x(0)})}{e(g^{r_i} . H(i)^{q_x(0)})} = e(g, g)^{rq_x(0)}$. $\perp$ denotes a special symbol returned by decryption.

Admission Control: Key management is important to control the membership of a group as in Fig. 3. There are many applications that require key management to control membership, e.g., video conferencing, collaborative workspaces, multi-party

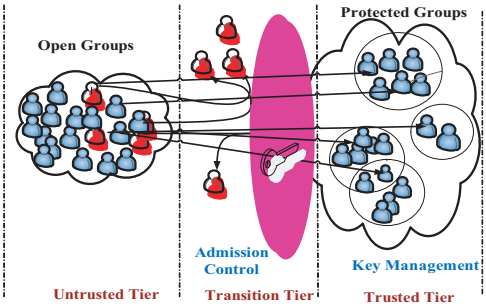


Fig. 3: Admission control and key management protected groups.

computing, digital broadcasting, software distribution, electronic learning, etc. The applications can take advantage of multicast technology. Peers can form open groups that have no restrictions for accessing the recommendations. In our case, groups are formed to safeguard the recommendations against attacks and other threats. Admission control is needed to allow only authorized users to join the group.

Keys do not have to be certified since we do not need to legally have authentication guarantees, as provided by certification authorities [32], [33]. Our approach follows a PGP-like strategy of distributing signed mappings and public keys via independent paths, and we apply a quorum-based strategy to find trusted mapping.

Key Generation: Let P be a set of n peers, q and k be non-negative integers with $k+1 \leq n$. A non-interactive *k*-secure, q-group key generation does the following: (i) Each group of q peers can compute the common key, for all $X \subseteq \{1,2,...,n\}$ with $|X| = q$, for all $u_X \in U_X$ with $i \in X$, a unique secret-key s_x exists such that for each peer, $peer_i$, it holds that $p(s_X|u_i) = 1$; (ii) Any group of *k* peers have no information on any key that they should not know. For all *Y*, with $|Y| = k$, $|X| = q$, and $X \cap Y = \emptyset$, for all *Y* and $X \subseteq \{1,2,...,n\}$, with $p_{u_Y}(u_Y) > 0$, and for all $s_X \in S_X$, it holds that $p(s_X|u_Y) = p_{s_X}(s_X)$.

The value held by $peer_{il}$, $l = 1,2,...,t$, and the identity of the other $q-1$ peers, has a unique value if a common key exists. The probability of a common key among peers $peer_{il},...,peer_{it}$ is s_X, where $X = \{i_1,...,i_t\}$ is the recommendations held by peers $peer_{j1},...,peer_{jk}$, where $Y = \{j_1,...,j_k\}$ and $X \cap Y = \emptyset$ which is equavalent to the prior probability that the common key is s_X.

V. The Proposed Approach

In this section, we propose a *gkeying* key generation approach to secure trust recommendations against threats from malicious peers in P2P e-commerce. We assume there are at most m groups and n peers in the P2P e-commerce environment $(m < n)$. Our *gkeying* approach is composed of six types of keys. We also assume peer links are bidirectional, i.e., if peer P can hear peer Q, then peer Q can also hear peer P. In our work, we consider a decentralized, distributed, and unmanaged P2P e-commerce with a large number of peers. The credibility and reliability required among the peers in a decentralized P2P e-commerce are questionable, due to anonymous and open nature of their existence to attacks. PGP has been well known for achieving these requirements through a web of trust. As time goes by, they grow into a web of trust that works well in a dynamic and decentralized environment [20].

Each peer advertises its trust information to others frequently and updates its recommendations among group members. In our work, we assume: 1) No centralized management authority; 2) Two peers represented by two different IP addresses are treated as different peers; 3) Peers can only see a part of the network.

Let $\{P_1, P_2, ..., P_n\}$ be a finite set of peers in P2P e-commerce. Every peer P_i has a unique identity that is non-transferable, bound for the entire lifetime, and verifiable. The identity of P_i is an *ID* chosen in different ways, i.e., 1) A device *ID* is bound to the hardware, e.g., the MAC address; 2) A network interface in case that the *ID* is derived from the IP address. We assume two peers are pairwise and securely connected by the Diffie-Hellman key exchange method. The peers at first have no prior knowledge of each other to jointly establish a shared secret. Recently, author [32] show that as long as each peer is able to obtain an updated verifiable secret sharing (VSS) information, there is no need for peer specific certificates.

In P2P, two peers have to evaluate each other in a group to carry out a trusted transaction where their secure relationship is administered by the pairwise key. Our P2P trust evaluation process considers two different trust components, namely, honesty and cooperativeness. The trust value that peer i evaluates toward peer j at time t, $T_{ij}(t)$ in (1) is represented as a real number in the range of [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. $T_{ij}(t)$, is computed by:

$$T_{ij}(t) = w_1 T_{ij}^{honesty}(t) + w_2 T_{ij}^{cooperativeness}(t), \quad (1)$$

where w_1 and w_2 denote the weights associated with the two trust components and $w_1 + w_2 = 1$. The P2P trust evaluation between two peers or two group leaders is conducted by evaluating one peer to the other, which is not symmetrical. This is when a peer i (trustor) evaluates a peer j (trustee) at time t and updates T_{ij}^X, where X indicates a trust component as illustrated in (2),

$$T_{ij}^X(t) = \begin{cases} (1-\alpha) T_{ij}^X(t-\Delta t) + \alpha T_{ij}^{X,direct}(t) \\ if\ i \text{ and } j \text{ are } one\text{hop neighbors}; \\ \underset{kEN_i}{\overset{avg}{}} \left\{ \gamma T_{ij}^X(t-\Delta t) + (1-\gamma) T_{kj}^{X,recom}(t) \right\}, \end{cases} \quad (2)$$

$\alpha T_{ij}^{X,direct}(t)$ indicates peer $i's$ trust level toward peer j based on direct observations accumulated over a time period, $[0,t]$, possibly with a higher priority given to recent interaction experiences over the time period $[t-\Delta t,\ t]$.

VI. Gkeying Approach and Attacks

In our approach, we propose *gkeying* that generates six types of keys. The keys generated ensure secure and trusted exchange of recommendation information among the peers within a group and also to other groups. We propose various

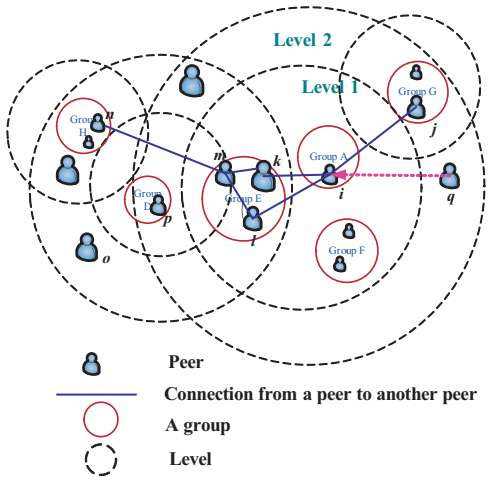


Fig. 4: Groups showing flooding with TTL.

ways in which malicious attacks can be minimized in grouped P2P e-commerce environments. In the approach, we use the neighbor similarity architecture [21] that enables peers to share common interest in a group. In a P2P e-commerce group infrastructure, peers communicate with others at different levels.

The connection adopts the small world network phenomenon with a characteristic path length in a long distance. At a random network, aggregation coefficient from a peer to another is high but the path length is small depending on the level as shown in Fig. 4. A peer is edged to its neighbors as per similar interest in a transaction for a particular service or goods. Optimal paths connect a peer to others in the same or different levels as shown in Fig. 4, i.e., $peer_i$, $peer_k$, $peer_m$, and $peer_n$ are connected. In Fig. 4, $peer_q$ establishes a connection to join group A. In establishing the connection, integrity of the peer has to be safeguarded from malicious attacks and other threats.

At each hop, TTL is reduced by one. If the neighbors do not have feedback for a peer, they forward the path discovery message to their neighbors until the TTL parameter is zero or a peer with the feedback is found. When a recommendation is not found, the source peer increases the TTL by one. In our work, TTL is defined by the levels shown in Fig. 4, where the TTL value is the maximum search depth.

A. Key Distribution in P2P E-Commerce

A key is a piece of input information for cryptography algorithms [2]. If a key is exposed, its encrypted recommendations will be disclosed. Before a peer in a group is initialized, investigation has been done to prove whether it is fit to join the group or not. If accepted, a key generation is done in which the peer is issued with a randomly generated initial key. The key is then broadcasted to all the legitimate peers who are members of the group.

A deterministic algorithm is used to decide the subset of keys to be allocated. In our approach, it is logical to find multiple paths between a pair of peers. In order to transmit a recommendation T_d to peer v securely, a peer u executes the following algorithmic steps:

$$u \rightarrow x : \{T_d\}_{k_{ux}}, x \rightarrow u : \{T_d\}_{k_{xv}}, u \rightarrow v : \{T_d\}_{k_{xv}}. \quad (3)$$

In (3), each peer maintains its own individual key and cooperates with others to generate the group key, which is also used for external linkages with other groups. A group is headed by a group leader (GL) who is a peer with added responsibilities. Creation of keys raises some fundamental issues of consideration: 1) Keys created require a pre-determined lifetime; 2) Key materials should be delivered in a secure manner to minimize potential threats; 3) It has to be protected using a verifiable and reliable mechanism; and 4) Protocols designed for group key management should protect against replay attacks and denial of service (DOS) attacks. Our approach combines pairwise keys with attribute set, in which we consider price, warranty, delivery, and availability online. The attribute set is based on the transactions of a peer and its neighbors. The method assumes that updates are given depending on trust evaluation of a peer by the neighbors' recommendations.

The work applies the "one-to-many" encryption scheme. ABE is a promising "one-to-many" encryption system, adopted in our *gkeying* approach. Handling of keys in multicast systems is complex because it operates in a dynamic environment. In unicast, key management mechanism can be typically implemented between two hosts in "one-to-many" neighbor relation. In ABE, policies will be associated with recommendations accumulated by peers. The encryption and decryption will be bound to the peers expected to receive the recommendations. An acknowledgement can also be bound with the same data. The attribute to be considered in our case is among neighbor peers during a transaction process.

B. Management of Keys to Safeguard Recommendations in Grouped P2P E-Commerce

In decentralized P2P e-commerce, management of keys is done in various levels in a group to ensure the trust and security of recommendation exchanged by the peers. In our case, *gkeying* supports the establishment of six types of keys, i.e., pairwise key, individual key, session key, group key, encryption key, and recommendation integrity key. Before peers form a group, the setup server randomly generates a bivariate t-degree polynomial [33], $f(x,y) = \sum_{i,j=0}^{t} a_{ij}x^i y^i$, over a finite field F_q, where q denotes a prime number. The prime number is large enough to accommodate a cryptographic key as it has the property of $f(x,y) = f(y,x)$. Both peer i and peer j can find the shared pairwise key with a single-variate polynomial $f(j,y)$. Both of them can compute the pairwise key $f(i,y)$ and store it in a peer i. Peer i can compute the pairwise key $f(i,j)$ and evaluate $f(i,y)$ at point i. From the property of symmetry of $f(x,y), f(i,j) = f(j,i)$, in which the pairwise key between peers i and j can be established.

Pairwise Key: Each peer interacts with its neighbors, by sharing a pairwise key with each of them. To obtain a pairwise key, peers generate n unique IDs $(n = m/p)$ $\{Id1, Id2, Id3, Id4, Id5, .., Idn\}$. They match each *PId* with m random *Ids*, $\{Id1 \rightarrow Id2, Id4\}$ $\{Id2 \rightarrow Id3, Id4\}$

$\{Id3 \rightarrow Id2, Id5\}$ $\{Id4 \rightarrow Id1, Id2\}$. A pairwise key for each pair of peers is generated and added to the key ring along with the *Id* key ring of $peer_1$ as $[K12|Id2]$ and $[K14|Id4]$.

The pairwise key between neighbor peers, encryption key, and T_dAC are generated by the peers themselves. This ensures that there is integrity, confidentiality, and authentication between one peer to another thus minimizes chances of malicious attacks.

Secret pairwise key between peer P and peer Q can be represented as K_{PQ}. Encryption is suitable for scenarios where an authorized peer outside the network needs to send a private query to a peer inside. For pairwise keys, the number of needed symmetric cryptographic keys is exressed as equation (4)

$$\binom{n}{2} = \frac{n(n-1)}{2} \tag{4}$$

for 1000 peers, there would be 499,500 keys. Each peer holds $n-1$ keys, and one for each possible communications.

When n becomes large, it is problematic or impossible to control the keys which is known as the n^2 problem. Given a key pool of size p and each peer is loaded with randomly selected m different keys from the key pool, the probability that two peers share at least one key is given as follows.

$$p1 = 1 - \left(\binom{p}{m} \binom{p-m}{m} \right) \binom{p}{m}^{-2} = 1 - \frac{((p-m)!)^2}{(p-2m)!p!}, \tag{5}$$

where $p1$ in (5) is the probability shared by the two peers. In summary, a pairwise key provides basic increased transmission and communication security, hence trust is enhanced. Each key illustrated as shown in (6) is unique and shared between two peers. For example, (P, Q) will share a key $K_{(P,Q),(1,Q)}$ with $(1, Q)$ and a different key $K_{(P,Q),(2,Q)}$ with $(2, Q)$, etc. Each peer stores $2(\sqrt{n}-1)$ keys and a total number of unique keys generated is $n(\sqrt{n}-1)$. If a unique pairwise key is shared by P and Q, and not shared by any other peer in the network, in case that P and Q are compromised, other links in the network are not affected. If P and Q share a pair of keys K_1 and K_2, the pairwise key between P and Q is calculated as:

$$K_{PQ} = K_{QP} = hash((K_1 \oplus K_2)||id_P||id_Q), id_P < id_Q. \tag{6}$$

We also note if a pair of keys K_1 and K_2 are present in three peers *P*, *Q*, and *R*, then the triple key is shown in (7)

$$K_{PQR} = hash\left((K_1 \oplus K_2) \| id_P \| \, id_Q \| \, id_R\right). \tag{7}$$

Individual Key: Each peer has a unique key shared pairwise with a group leader(GL). The key is used by the GL to authenticate the peers. The key is incorporated to the recommendations. The peer contributes directly to produce the individual key, as well as the GL.

Session Key: This is a global key shared by all peers in a group. Each peer uses a combination of an individual key and a session key. We propose the use of a distributed scheme to generate a session key among the peers. Then, the session key can be associated with the group key and dynamically distributed among the peers. The probability that two peers with the same number of random keys share at least one key is shown in equation (5).

When attempting to determine whether or not a key is shared between a pair of neighbor peers, the peers broadcast a plaintext listing their key identifiers. The identifiers are randomly assigned to each of the peers in a group, hence do not give attackers any additional information about the key values. If a neighbor has a key corresponding to one of the broadcasted identifiers, it answers the source peer with a challenge-response message. Peers who do not directly share keys can establish pairwise keys through commonly trusted neighbors. For secure trust concern, groups require an access control mechanism to authorize where the member peers can access the group privileged services. Access control is usually achieved by encrypting the content using the session key (SK) shared by all legitimate group members. The group membership is dynamic and the encryption key is updated to prevent the leaving/joining peer from accessing the future or prior communications. This ensures that malicious peers cannot use previous keys to launch attacks.

For a group with N members or peers, the length of an ID is $n = \log N$ and the total number of bit assignments is $2n$. Two binary values are mapped to one bit position (one for value 0, and one for value 1).

Group Key: The peers in a group identify themselves to the group with a key, referred to as a group key. The key is used to authenticate members and encrypt recommendations. Each peer in the group will collaboratively contribute its part to the group key. Modification of membership requires the group key to be refreshed to ensure backward and forward secrecy. Whenever group membership changes, a new group $P_y = \{U_1, ..., U_n\}$ is formed and its members establish a confidential channel through an instance performing a group key agreement protocol (GKA). The major goal of the GKA protocol is to establish a confidential channel for member peers in a group. For a given peer v, we associate a secret (or private) key known as the individual key K_v, and a blinded (or public) key BK_v. All arithmetic operations are performed in a cyclic group of prime order p with the generator α. Therefore, the blinded key of peer v can be generated by $BK_v = \alpha^{Kv} \bmod p$.

In our approach, multiple entities are responsible for managing the group as opposed to a single entity, and any alteration can be detected easily as each peer stores information of others in the group. Dynamic groups in P2P e-commerce key management is a difficult problem because of the requirement of scalability under the restrictions of available resources and unpredictable mobility. A group key protocol allows a set of users to communicate over an open network and agree on a private session key. Group members merely negotiate a common encryption key (accessible to attackers), but hold respective secret decryption keys. The group key is denoted by K_m^G peers in a group while S_i must have $\{K_m^G, \forall m : t_m^i = 1\}$.

In e-commerce applications, peers switch between different groups by subscribing or unsubscribing. We introduce the notation $A_i \rightarrow A_j$, which represents a peer switching from group A_i to group A_j, i.e., peer join $(A_0 \rightarrow A_i)$ and peer departure $(A_i \rightarrow A_0)$. The rekeying messages are transmitted

when one peer switches from A_i to A_j represented by C_{ij}. Switching from group A_i to group A_j is equivalent to adding the subscription to the group: $G_m, \forall m : t_m^i = 0$ and $t_m^j = 1$. To drop subscription of a group using the tree-based key management scheme, the rekeying message size is calculated as follows.

$$C_{ij}^{ind} = \sum_{m=1}^{M} \max(t_m^i - t_m^j, 0) * (d.f_d(n(G_m))). \quad (8)$$

It is noted $\max(t_m^i - t_m^j, 0)$ in equation (8) equals to 1, when $t_m^i = 1$ and $t_m^i = 0$. Therefore, when the term equals to 1, $d.f_d(n(G_m)$ rekeying messages are necessary to update keys on the key tree associated with the group. We conclude that key management occurs when a peer switches from A_i to A_j, where $i \neq j$. When a user switches from group A_i to group A_j, it is necessary to: 1) Update the group keys of a peer not to access the previous information in the group, and 2) Update the group key of a switching peer not to access the future communications in the group. A group key protects multiple peers at the same time, and is able to scale from a small to a large number of peers. Each peer stores an individual key, the session key, and a set of encryption keys.

Most group key agreement schemes authenticate members using certificates and PKI. The admission control framework proposed by [13] is also based on PKI and certificate. Disadvantages arise: 1) The certificates need to be exchanged, which consumes bandwidth because of the large size, and 2) The signature of the PKI needs to be generated and verified which is computationally expensive. Consequently, we use a group key that is combined with other keys. An efficient key revocation takes place upon the revocation time specified per group. It happens when a peer misbehaves or other peers vote to remove the peer from being the leader. In case that a peer leaves the group, all the shared keys are revoked and updated. The group key is generated in a shared and contributory fashion, hence there is an increase in system reliability and no single-point-of-failure. A group leader uses a group key to encrypt messages and broadcast them to the group members, so that only the group members can use the group key to decrypt the ciphertext message.

Encryption and Recommendation Integrity Key: Recommendation integrity involves two aspects: 1) Source integrity that verifies the identity of the source, prevents the acceptance of messages, and neighbor recommendations from a fraudulent source, and 2) Recommendation integrity that prevents modification.

Recommendation integrity key (T_dAC) is a small fixed-size block of data that is generated based on a recommendation T_d of variable length and the secret key K. It can be referred to as cryptographic checksum expressed as $T_dAC = C(K, T_d)$. If P wishes to send a recommendation T_d to Q and protects it via a T_dAC, they first need to share a secret key K. In addition, P calculates code T_dAC as a function of T_d and K. Then, the recommendation T_d plus the code T_dAC are transmitted to Q. Q performs the same calculation on T_d using K to generate a new code T_dAC'. The received code T_dAC is compared with the calculated code T_dAC' to verify the integrity.

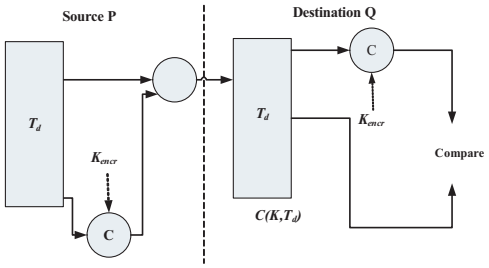


Fig. 5: Recommendation authentication.

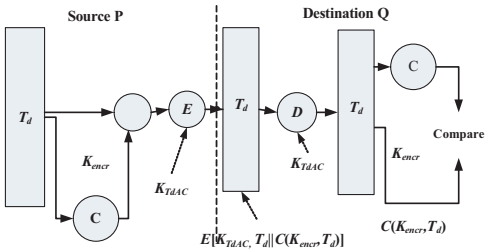


Fig. 6: Recommendation authentication, confidentiality and integrity by using recommendation encryption.

In P2P e-commerce, peers have recommendations that are constantly propagated across other peers. We use encryption key K_{encr} and recommendation integrity key T_dAC to ensure the security of a message as shown in Fig. 5 and Fig. 6. Let P and Q be two entities in a peer group, K_{PQ} being the pairwise key shared by P and Q, and W be the shared information between P and Q. Let the length of F (pseudo-random number) be λ. We compute the key for encryption and for authentication of length λ by applying F repeatedly on K_{PQ} to obtain keys as $\lfloor K_{encr}||K_{T_dAC} \rfloor$. Therefore,

$$K_{encr} = F(K||1||W), \quad (9)$$

and

$$K_{T_dAC} = F(K||2||W). \quad (10)$$

The complete message that *P* sends to *Q* is:

$$P \rightarrow Q : \{T_d|T_s\}_{K_{encr}}, T_dAC(K_{T_dAC}, \{T_d|T_s\}_{K_{encr}}) \quad (11)$$

where T_d denotes a recommendation and T_s denotes the timestamp when sending the recommendation. The T_dAC value protects both a recommendation integrity as well as its authenticity by allowing verifiers to detect any changes to the recommendation. The T_dAC is different from digital signature as the T_dAC values are generated and verified using the same secret key. In summary, considering peer A and peer B in the group, keys are summarized in Table 1.

C. Peer Leaving

The identity of each peer is of n-bit length, where $n = \log N$. A group regulator (GR) is responsible for key generation and distribution, and the group data is encrypted by a SEK. It generates and distributes a set of n secrets, which are one-to-one mapped to the bits in $p_i's$ ID. If a peer p_i is removed, a GR will update $\{\lambda, SEK\}$ group key and send the new key including the information of the revoked peer to all

TABLE I: Summary of the types of keys in *gkeying* and their function

Key	Entities	Function
Individual key	Peer A and peer B	Between two peers
Pairwise Key	Peer A and group leader	Between a peer and the group leader
Session Key	For all peers	For all peers during a session
Encryption Key	To encrypt data	Encrypt data in the group
T_dAC	Peer A and peer B	To authenticate data in a group
Group Key	Between one group leader to another	To authenticate a transaction in a group

other peers in the group through a secure and trusted channel. This can be done as follows,
1) The GR first changes

$$\lambda' = \{\beta, g^{\alpha\prime}, e(g,g)^{\alpha'}\}, \tag{12}$$

where α' is selected randomly in $\mathbb{Z}_p$.
2) α and β are non-trivial random numbers.
3) GR multicasts an encrypted key-update factor,

$$ku_f = g^{\frac{\alpha'-\alpha}{\beta}}. \tag{13}$$

Note that ku_f is encrypted and can not be decrypted.
4) Each peer p_i updates its private key SK_u based on the key updating factor $g^{\frac{\alpha'-\alpha}{\beta}}$.
Only the remaining peers can recover the message and update SEK as well as their private keys. Moreover, to support dynamic group changes, SEK needs to be updated when group members join or leave the multicast group to ensure backward secrecy and forward secrecy. A peer may unsubscribe from a group without broadcasting information to the others. Peers that exhibit such behaviors are called faulty peers and exhibit malicious intent.

Considering the dynamics of P2P e-commerce, peers should update their neighbor peers regularly. A peer can change interest due to the availability of new products and business opportunities. Chord uses a stabilization protocol [35] to regulate the dynamically formed topology according to varying interest. Some peers can exhibit a dynamic personality, i.e., switching between a honest behavior and a dishonest behavior. *Reputation milkers* or *oscillating peers* are one type of peer personality that builds a good reputation and then takes advantage of it to launch attacks. When a peer P leaves the network, its DHT table will be taken over by the closest neighbor P'. In order to deal with an abrupt departure, P' should cache the information kept of P.

D. Key Revocation

We consider two situations in the key revocation process in P2P e-commerce: 1) The compromise of a group key, and 2) The compromise of an individual key. Identifying the peer who has duplicated recommendation, has led to key revocation. A neighborhood peer can identify and report, then verification is done to confirm who is the genuine peer or the Sybil attack peer. P2P e-commerce is decentralized and force each peer maintaining a revocation list, which include identity of peers that have been revoked. When a peer is compromised due to numerous attacks, the peer can be removed from the group. After the compromised key is added to the revocation list, the peer is not able to recover the key anymore. To prevent keys to be compromised, the authentication authority sets a limit on lifespan of each peer's individial key to use the key material.

Consequently, using long-term keys has the possibility that some of the keys may be compromised before their normal expiry time, therefore they need to be updated or revoked. We distinguish two cases: 1) A peer's device is lost, stolen, or damaged. In case the entire peers database is compromised, each peer needs to contact the authentication service in which the group leader has to acquire new credentials (key update). Furthermore, the peer's old credentials need to be rendered unusable, so that no other party can impersonate the peer (revocation); 2) If a peer quits or misuses the service, the authentication service needs to render the peer's keys unusable (revocation). The revocation mechanism is based on group key revocation list. In our case, the GL keeps a list of indices of all peers whose keys have been revoked.

The peers need a mechanism to verify their integrity. This can be done by each peer computing an integrity code in conjunction with the GL. The ID can be associated to an expiration date for key revocation purposes. In this case, the GL needs to check that there is no peer with duplicated ID and different expired dates to prevent ID impersonation.

To achieve dynamic revocation in a P2P e-commerce, if a peer detects that another peer is malicious, it sends a revocation notification that includes the malicious peer and itself considering that its own life is less important than the goodness of the group. However, the mechanism has limitations. It can only be used within our proposed approach because in the group all peers have the same interest.

In our approach, we consider two situations in the key revocation process: 1) The compromise of a group leader (GL), and 2) The compromise of a peer. We assume that each peer maintains a peer revocation list (PRL). Only valid entities are involved in the network. Case 1: When a peer is compromised, it can be removed from the session where all the keys are updated. After the compromised peer identifier is added to the revocation list, the revoked peer can not recover the new session key hence cannot reveal the new encryption keys and the recommendation integrity key. Case 2: When a GL is compromised, it needs to be removed from the group. The compromised group key can be removed from the group not to reveal the next session key. To do this, the compromised group key will be forced to leave the group and a new group key is generated. When a GL and a peer are compromised, the GL will be removed from the group, preventing it from launching attacks.

The group members should recover the keys on their own. In our case, they will send broadcast messages to the GL. The group leader's broadcast message should be bi-directional. A

primary challenge is how to provide an acceptable security. Non authenticated group members have to be prevented from participating in the group activities. Information transmitted within the group at a certain session should be safeguarded securely through encryption by a session key. In key healing, particular evaluation measurements should be done, i.e., forward and backward secrecy.

E. Efficiency of Malicious Behavior on Recommendations

Reputation systems help buyers to decide whether to purchase a product or not. Without an efficient reputation system and key management system, a reasonable number of such malicious peers can collude to rate good peers badly. Our work has proposed ways to manage keys to avoid retaliation and minimize fake ratings due to collusion. In our proposed model, trust reports should remain encrypted and not open during the transmission processes, hence guarantee secure trust. When peers join the group, they acquire different identities to be part of the group. Each neighbor is connected to the peers by the success of the transaction it makes or the trust evaluation level. In our approach, we consider how two peers can be controlled by a malicious peer. If peer i tells peer j its secrecy, then peer j can masquerade as peer i to all of peer j neighbors with whom peer i shares a pairwise key, and vice versa. The keys from each successively obtained peer can be reused by the other attacker-controlled peers, cascading the impact of peer compromise. Each peer is evaluated on its trustworthiness based on recommendations by the neighbors. Peers can be identified to be Sybil attacck peers and collusion attack peers. In our work, collusion can also be associated with the property of backward secrecy and forward secrecy. A peer can still collude with others who have left the group to launch attacks. Keys, which link a peer to others, like the pairwise key, have to be updated in all members of the group. If a peer p declares that it received some service from peer q, it is desirable to have a proof that it happened, which would prevent the collusion between the two peers.

As keys changingin different sessions, the groups can eliminate any association of malicious peers. Our approach ensures that dynamism is maintained, where a peer ensures the interests of the group are to be safeguarded more than its own. An edge $i \rightarrow j$ represents a peer relationship whereby two peers communicate. In the case that the information is encrypted and decrypted, a comparison is made to identify whether the information is the same. Such key pools are vulnerable to a colluding minority of attacker-controlled peers.

VII. Security and Performance Analysis

A. Security Analysis

In this section, we make an analysis on trust and security of the approach. The *gkeying* satisfies the following characteristics:

1. Only the authorised peers can communicate in the network. K_{encr} and T_dAC ensures there is secure communication.
2. The distribution of session keys among the peers is secure.
3. Compromised peers are isolated by the other peers.

Malicious peers try to attack the session key, but this fails because the session key is just for a short period in our approach. We propose that when a peer is compromised, the key material stored in the peer will be extracted by the adversary. The key extracted will be used to attack the network hence has to be revoked immediately.

Let *K(N, O, d)* be the expected value of the rekeying and revocation. It is used in removing N peers from the group. There are d^l key encrypted keys (KEKs) at the l^{th} level of the group connected to the peer as neighbors. For $L = 0, ..., R-2$ and $L = \lceil \log_d O \rceil$ number of KEKs at the $(L-1)^{th}$ level:

$$S_1 = \left\lceil \frac{O - d^{L-1}}{d-1} \right\rceil . \tag{14}$$

Let α^l be the number of KEKs to be updated at the level l, if N peers leave the group. The expected value is expressed as:

$$K(N, O, d) = E\left[\sum_{l=0}^{L-1} \alpha_l\right] = \sum_{l=0}^{L-1} E\left[\alpha_l\right]. \tag{15}$$

The session key distribution process is secure as it is based on the individual keys [36] and any activity that takes place. In the *gkeying* approach described, the revoked peer can not recover the session and the group key. At the same time, an outside attacker can not masquerade as a peer in the group disseminating a session key and start a revocation attack either. In our proposed approach, only the authorized peer can communicate in the group. The system ensures secure trust and integrity, as peers outside the group can not communicate without being assigned the neccessary key materials. In the approach, the key cracking time is more than the session key time, which means it will be a waste of time to crack when the key will have already expired and renewed.

We also evaluate trust and security of our approach in terms of attacks and threats. Malicious attacks in P2P e-commerce can be classified into outsider attacks and insider attacks. While most outsider attacks can be prevented by authentication and cryptography, insider attacks are much harder to deal with. With P2P e-commerce trust evaluations reported from peers, a group leader obtains a comprehensive trust report toward all peers in its group and can perform statistical analysis to identify and exclude malicious peers in the network.

In *gkeying*, peers are assured of each other's identities by possessing the appropriate pairwise key. Eavesdropping attack can arise, which is addressed by encrypting the pairwise key. Each peer stores a random set of s*r pairwise keys, whereby a peer x can reach a set of peers $N(x)$. Each peer in a group can contact other peers. Peers broadcast their identifiers to neighbors, who examine their IDs to determine if they share a pairwise key. For example, if a pairwise key $K_{m,x}$ is added to $U(m)$, the number of usable pairwise keys is m, if one of peer m's neighbors $x \in N(m)$ holds K_{mx}. If a peer is captured, the trust, integrity and availability of the peer is under threat. Attacker-controlled peers increase their chances of partitioning the group and counteract redundant routing.

In our analysis we consider a peer x who has an individual key K_x, which has been compromised by adversary τ. At first, peer x chooses a random $t_x \in Z_q^*$, computes $T_x = t_x W$, and

sends a broadcast message (ID_x, l_x, R_x, T_x). τ intercepts the message and attempts to derive z_1, z_2, z_3 so as to compute the pairwise key. τ pick a random number t'_y and calculates $T'_y = t'_y W$, $z_2 = t'_y(R_x + H(W_{pub}, id_x, l_x, R_x))W_{pub}$, $z_3 = t'_y T_x$. τ is not able to compute Z_1 from equation $z_1 = S_y T_x$ as it doesn't know S_y, hence the attack fails. Our approach provides perfect forward secrecy. This is because the adversary τ who compromises the peer x and obtains $x\prime s$ key $K_x = (R_x, S_x)$, is infeasible to reveal previous established key $K_{x,y}$ between x and neighboring peer y, even if τ compromises the peer y.

B. Performance Analysis

In this section, we evaluate the key generation indices of our approach and compare it with other approaches. In [37], security analysis of an attacker can not determine non-compromised keys established with a polynomial, if more than t compromised peers have shares of the polynomial [38]. In our *gkeying* approach, it is easier to isolate, revoke, and expel any of the misbehaving peers from the group. Many of the schemes investigated employed the pre-key distribution, which is not applicable to P2P e-commerce due to anonymity characteristics.

Computation Cost: There are six types of keys in our proposed approach. To calculate the individual key, pairwise key, session key, and group key, a polynomial is used. The *gkeying* approach is efficient in the computation of polynomial evaluation. The calculation of the encryption key and the recommendation integrity key is based on a pseudo-random function (PRF). PRF is a deterministic function $f: \{0,1\}^n \rightarrow \{0,1\}^n$, which is efficient and takes two inputs x and $k \in \{0,1\}^n$. We consider x to be a variable and let k be a hidden random seed function index $f(x,k) = f_k(x)$. The *gkeying* approach is efficient in computation, generation of public key and private key used in the encryption and decryption. A public key encryption is a triple of algorithms $E = (G; E; D)$, where: a) G is the key generation algorithm in which $G(I^k)$ outputs $(PK; SK; M_k)$, SK denotes the secret key, PK is the public key, and M_k is the recommendation space associated with the PK/SK-pair, where k is the security parameter, which determines the security level; b) E is the encryption algorithm for any $m \in M_k$, and c is the cipher text; c) D is the decryption algorithm, $D(c; SK) \rightarrow m \in \{invalid\} \cup M$ is called the decrypted message.

Communication Cost: The key is sent with the recommendation feedback from one peer to another. The session key is distributed to the network by directed flooding from one peer to another based on individual key recognised in the group. If a peer is compromised, the information is broadcasted to all peers as key revocation is being done.

Two peers need to establish an indirect key in two ways: 1) Either the two peers are in the same deployment group, or 2) The key establishment involves peers in the deployment group.

If the two peers are in different deployment groups, the path key establishment involves those in the same deployment group with the source peer or the destination peer. In addition, we note if two peers in two deployment groups are neighbors, then corresponding deployment groups have high probability of being close to each other. This can reduce the overall communication overhead significantly during their key generation.

Storage Cost: Let d represent the number of neighboring peers to a particular peer in a group. Each peer needs d storage units for the pairwise keys, m storage units for the individual peers, and four other storage units for the session key, group key, encryption key, and the T_dAC key. To calculate the pairwise key and the individual key, a peer needs $t+1$ storage units for the t-degree polynomial whereby each peer is loaded with m secret shares to recover the session key. The total keys required for each peer is $2m + d + t + 5$, hence the peer has to be loaded with secret shares to recover the session key. The more the number of storage peers, the less the size of shares per storage. The increase in the cost of storage is as a result of keeping records of its neighbors.

Our approach has a better guarantee of secure trust in all interactions based on common interest, where peers become acquainted to each other. Peers need to have same interest to trade on common products in which their recommendation trust is guaranteed by the six types of keys. The keys are safeguarded by all the peer members in the group. Our work is better than the work of [39] and [40], whose open problem is to incorporate certificate revocation and expiration into P2P-PKI. We have different keys for different purposes in the group, which guarantee secure and trust in P2P e-commerce transactions. Communication overhead measured by the rekeying-message-size is a major performance criteria for key management schemes [34].

In summary, we evaluate the performance of *gkeying* approach as shown by the experimental results. All the existing key distribution guarantees 100% key connectivity with different storage cost. The security analysis has been done for the group leaders and the peers, where the number of group leaders and peer nodes in a P2P e-commerce is m and n respectively. It is a practical assumption that $m << n$. When the peers are randomly and uniformly deployed in the P2P e-commerce infrastructure, there are $\lceil n/m \rceil$ peers in each group. Each GL can keep one hop communication with at most $\lceil m/3 \rceil$ group leaders in different groups. A group leader links the group to other neighbor groups with $\lceil m/3 \rceil$ group leaders. Each GL stores a symmetric master key K_m, a public/private key pair and public keys of the group leader and the other group leaders. The neighboring group leaders can securely establish a symmetric key with each other.

In the *gkeying* approach, there are six types of keys, i.e., individual key, pairwise key, session key, group key, encryption key, and recommendation integrity key as discussed in earlier sections. If we assume n is the number of neighbors to a particular peer, each peer will have $(n+3)$ keys. N is the total number of keys in the group. Each GL will store $N+2+\lceil m/3 \rceil$ while a group has $\lceil N/n \rceil$ peers. With different value of N, we obtain Table II and Fig. 7. In Table II, we give N different values, i.e., $N1$, $N2$, $N3$, and $N4$. The values of m are 1 to 10. In a particular session $1 \leq x \leq m$, the GL computes $Rw_1 = \sum_{i=1}^{n} R_i$ as well as $(|\lambda_j| - 1) \times |\lambda_j|$.

TABLE II: Distribution of generated keys

m	$N1$	$N2$	$N3$	$N4$
1	12	22	32	42
2	13	23	33	43
3	13	23	33	43
4	13	23	33	43
5	14	24	34	44
6	14	24	34	44
7	14	24	34	44
8	15	25	35	45
9	15	25	35	45
10	15	25	35	45

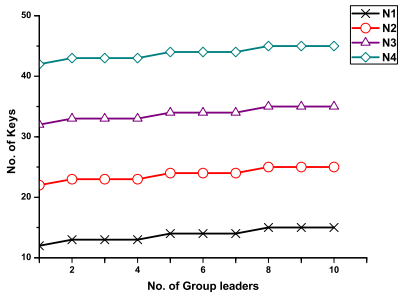


Fig. 7: Key space reguirements of the *gkeying* key generation approach.

C. Maliciouness by uncertain Peers

We evaluate the performance of our system in suppressing dishonest interactions. With uncertain peers in the group, the percentage of malicious peers varies from 10% to 80% without any malicious peer participating in our approach. In our simulation we used a network of 100 honest peers, half of which are organized in a group where six keys are generated. In our simulation, peers to commit transaction are selected randomly from honest peers. The bars in our results in Fig.8 show the fraction of bad interactions in the same period of time. We note from the experiment the percentage of bad interactions increases with the increase of the number of peers. From the comparison of the two experiments, in our work and [8] we note that the our approach has a lower percentage of bad interactions, which means it is more secure and has a higher trust level.

D. Compromised Peers

In group P2P e-commerce, each group has a total of N keys. Each peer has an individual key connected to its neighbors. The probability that a key doesn't belong to a peer is given by $\frac{N-w}{N}$. If there are n compromised peers in the group, equation (16) shows the probability that a given key is not compromised.

$$\left(\frac{N-w}{N}\right)^n. \tag{16}$$

The probability of compromised keys in a group is shown in equation (17)

$$p = 1 - \left(\frac{N-w}{N}\right)^n. \tag{17}$$

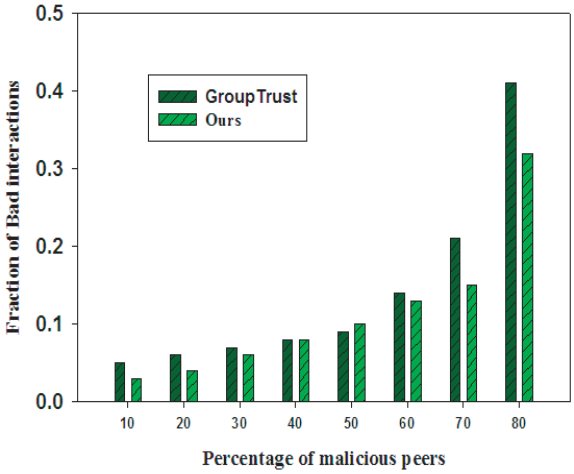


Fig. 8: Peers in a group with uncertain peers.

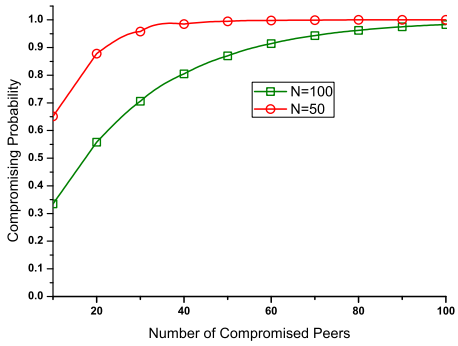


Fig. 9: Compromising probability.

When a key is compromised, the pairwise key it shares with the group leader and other peers are disclosed. Two peers x and y have a commitment as they share the pairwise key. It depends on the probability that a peer has only one common commitment with its neighbors. Each peer has d neighbors and the probability each peer has only one common commitment with its neighbor is μ. Probability a key is not disclosed by a compromised peer is $\frac{N-\mu*d}{N}$. If there are n compromised peers, the probability a peer is not compromised is

$$p = \left(\frac{N-\mu*d}{N}\right)^n \tag{18}$$

Therefore, probability of total compromised keys is

$$p = 1 - \left(\frac{N-\mu*d}{N}\right)^n. \tag{19}$$

The probability of the total number of compromised keys, where n number of peers are malicious is shown by equation (16) and illustrated by Fig.8. Each group has a total of 6 keys and each peer can have at least 4 keys. From Fig. 9, we can see that with increase of compromised peers, the compromised probability increases. With more neighbors, the peer compromising probability is less. This is because when neighbors are many in a group, there is more monitoring and validation.

Our *gkeying* approach is better than the work in [40], whereby the authors developed a specialized P2P-PKI realizing efficient search and transfer of certificates and trust recommendations. Their work was based on logic calculus. We are able to control and monitor the operations of a peer in a group and also link to other groups in a secure trusted and efficient manner. The proposed *gkeying* ensure reliable transfer of recommendations from one peer to another. The recommendations delivered to the requesting peer can be decrypted and compared to the original recommendations.

The proposed approach can be implemented at the formation stage of groups by traders dealing with common goods and services. This will increase the confidence of business enterprises in certain regions. The P2P e-commerce transactions will be guided by particular constitutional laws. In case anything happens in a transaction, the laws can be applied to deal with the malicious peer who committed the e-commerce transaction offence. It can also deal with the traders who take advantage of their social and economic ties to fraud the business by compromising the actions of others. Our approach is able to identify maliciouness faster compared to [8], which only relies on interactions of the peers during transactions. Credibility as a source factor ensures that each peer keeps a good relationship with others for the sake of the group with help of the six types of keys. If the source is trusted, it is more likely that the P2P e-commerce transaction will be successful.

To secure the link other than the traditional means of safeguarding transactions, the trust of the concerned peer is a crucial issue that needs to be evaluated. In our approach, the decentralized transaction is guided by factors, such as proximity and personal preference. The groups formed, in addition to six types of keys, ensures trust in exchanged information. The system has a verification procedure after any information delivered, as shown by the comparison of the encrypted and decrypted information. Peers with multiple identities are identified from the group, which reduces Sybil attacks. In other words, it serves to thwart the ability of compromised peers to collude and disrupt P2P e-commerce transactions. In this work, we establish a secure routing structure over which messages and data can reliably be exchanged in the presence of malicious peers and other threats. The paper improves the quality of services in P2P e-commerce transactions.

VIII. Conclusion and Future Work

We proposed the *gkeying* approach, which aims at the generation of six types of keys and the special encryption of recommendations being transacted in P2P e-commerce. Our analysis shows that the proposed approach has trust and secure against the compromise of peers in P2P e-commerce. Our approach combines key management and secret sharing where the system secret is distributed to a group of peers. It is easier to coordinate trust and security in a group rather than the entire network, which means our approach provides more integrity, control, and reliability. It caters the dynamic behavior of peers whereby the update of keys is a continuous process. This work is of interest to the trust and security community as it brings a new idea of monitoring and eliminating malicious behaviors in P2P e-commerce transactions.

Further investigations include security related to the recommendation acquisition and the routing issues. More research needs to be done on how to distribute keys if a peer is a member of many groups and trade randomly with non-group members.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grant numbers 61073037 and 61103035, and the Ministry of Education Fund for Doctoral Disciplines in Higher Education under grant number 20110162110043.

References

[1] H. Yu, M. Kaminsky, and A. Flaxman,"SybilGuard: Defending Against Sybil Attacks via Social Networks," *Proc. of SIGCOMM 2006*, September 2006, pp.267-278.

[2] B. Wu, J. Wu, E. B. Fernandez, and S. Magliveras, "Secure and Efficient Key Management in Mobile Ad Hoc Wireless Networks," *Journal of Network and Computer Applications*, vol. 30, no. 3, August 2007, pp.937-954.

[3] S. H. Al-Bakri, M. L. M. Kiah, A. A. Zaidan, B. B. Zaidan, and G. M. Alam, "Securing Peer-to-Peer Mobile Communications Using Public Key Cryptography: New Security Strategy," *International Journal of Physical Sciences*, vol. 6, no. 4, February 2011, pp.930-938.

[4] Y. R. Chen, J. D. Tygar, and W.G. Tzeng, "Secure Group Key Management Using Uni-Directional Proxy Re-Encryption Schemes," *Proc. of IEEE INFOCOM 2011*, April 2011, pp.1952-1960.

[5] L. Ertaul, N. Chavan, "Security of Ad Hoc Networks and Threshold Cryptography," *IEEE 2005*, pp.1-6.

[6] S. Capkun, L. Buttyan, and J. P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, March 2003, pp.52-64.

[7] S. Capkun, J. P. Hubaux, and L. Buttyan, "Mobility Helps Peer-to-Peer Security," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, March 2003, pp.43-51.

[8] Y. Zhang, H. Zheng, Y. Liu, K. Li, and W. Qu, "A Group Trust Model Based on Service Similarity Evaluation in P2P Networks," *International Journal of Intelligent Systems*, vol. 26, no. 2, 2011, pp.47-62.

[9] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *Special Issue on Network Security, IEEE Networks*, December 1999, pp.24-30.

[10] G. Kambourakis, E. Konstantinou,and S. Gritzalis, "Revisiting WiMAX MBS security," *Computers and Mathematics with Applications,Elsevier,*60, 2010, pp.217-223.

[11] H. Chan, A. Perring, and D. Song, "Random Key Predistribution Scheme for Sensor Networks," *Proc. of IEEE Symposium on Security*, May 2003, pp.197 - 213.

[12] G. Wang, Q. Du, W. Zhou, and Q. Liu, "A Scalable Encryption Scheme for Multi-Privileged Group Communications," *Journal of Supercomputing, Springer,* Published online on September, 2011, DOI: 10.1007/s11227-011-0683-4.

[13] Y. Kim, D. Mazzocchi, and G. Tsudik, "Admission Control in Peer Groups," *Proc. of IEEE International Symposium on Network Computing and Applications*, 2003, pp.309-329.

[14] Y. Sun and K. J. R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," *Proc. of IEEE INFOCOM 2004*, 2004, pp.1296-1306.

[15] L. Eschenauer and V. Gligor. "A Key-Management Scheme for Distributed Sensor Networks," *Proc. ACM CCS*, 2002, pp.41-47.

[16] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation Based Trust for Peer-to-Peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, July 2004, pp.843-857.

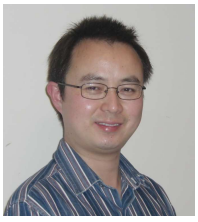
[17] A. Josang, R. Ismail and C. Boyd, "A Survery of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43, no.2, 2007, pp.618-644.

[18] T. D. Subash, C. Divya, "Novel Key Pre-Distribution Scheme in Wireless Sensor Network," *Proc. of ICETECT*, March 2011, pp.959-970.

[19] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks," *Proc. of ACM Conference on Computer and Communications Security, CCS, Washington, DC, USA,* October 2003, pp.42-51.
[20] Z. Xu and H. Jiang, "Framework of Decentralized PKI Key Management Based on Dynamic Trust," *Proc. of IPCV,* 2008, pp.1-7.
[21] F. Musau, G. Wang, and M. B. Abdullahi, "Group Formation with Neighbor Similarity Trust in P2P E-Commerce," *Proc. Joint Conference of IEEE TrustCom/IEEE ICESS/FCST,* November 2011, pp.835-840.
[22] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, "A Survey on Cluster-Based Group Key Agreement Protocols for WSNs " *IEEE ommunications Surveys and Tutorials,* vol. 13, no. 3, THIRD QUARTER 2011, pp.429-442.
[23] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. D. Ferrer, "Asymmetric Group Key Agreement," *Advances in Cryptology - EUROCRYPT, Lecture Notes in Computer Science (5479), Springer-Verlag,* 2009, pp.153-170.
[24] Z. Zhou and D. Huang, "Constructing Efficient Attribute-Based Broadcast Encryption," *Proc. of IEEE INFOCOM 2010 on Computer Communications Workshops,* March 2010, pp.1-2.
[25] D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. of Computing,* vol. 32, no. 3, 2003, pp.586-615.
[26] V. Goyal, A. Pandey, A. Sahai, and B. Waters, "Attirbute-Based Encryption for Fine Grained Access Control of Encrypted data," *Proc. of ACM Conference on Computer and Communicatioins Security, CCS,* 2006, pp.89-98.
[27] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," *Proc. of IEEE Symposium on Security and Privacy,* May 2007, pp.321-324.
[28] K. Hoeper and G. Gong, "Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes with Key Revocation," *Technical Report, University of Waterloo, Canada,* 2006.
[29] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. "Reputation and Trust-Based System for Ad-Hoc and Sensor Networks," *In Algorithms and Protocols for Wireless Ad-Hoc and Sensor Networks,* A. Boukerche (ed.), Wiley and Sons, 2008.
[30] J. Liao and Y. Ishikawa, "A New Concurrent Checkpoint Mechanism for Real-Time and Interactive Processes," *Proc. of IEEE Computer Software and Applications, COMPSAC,* July 2010, pp.47 - 52.
[31] S. M. R. Farshchi, F. Gharib, R. Ziyaee "Study of Security Issues on Traditional and New Generation of E-Commerce Model," *Proc. of IPCSIT (9), IACSIT Press, Singapore,* 2011, pp.113-117.
[32] N. Saxena, G. Tsudik, J. H. Yi, "Efficient Node Admission for Short-Lived Mobile Ad Hoc Networks," *Proc. of Networking Protocols, ICNP,* Nov 2005, pp.268-278.
[33] Y. Wang, B. Ramamurthy, and Y. Xue , "A Key Management Protocol for Wireless Sensor Networks with Multiple Base Stations," *Proc. of ICC, IEEE,* 2008, pp.1625-1629.
[34] M.J. Moyer, J.R. Rao, and P. Rohatgi, "A Survey of Security Issues in Multicast Communications," *Proc. of IEEE Network,* vol. 13, no. 6, December 1999, pp.12-23.
[35] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balarikishnan, "Chord: A Scalable Peer-to-Peer Lookup Service For Internet Applications," *Proc. of SIGCOMM, ACM, San Deigo, USA,* August 2001, pp.149-160.
[36] D. Liu, P. Ning, and K. Sun, "Efficient Self-Healing Group Key Distribution with Revocation Capability," *Proc. of CCS, NewYork, NY, USA: ACM Press,* 2003, pp.231-240.
[37] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U .Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences, " *Springer,* 1998, pp.471-486.
[38] S. Ruj, A. Nayak, and I. Stojmenovic, "Fully Secure Pairwise and Triple Key Distribution in Wireless Sensor Networks Using Combinatorial Designs," *Proc. Mini-Conference of IEEE INFOCOM 2011,* 2011, pp.326-330.
[39] H. Jiang, R. Zhang, Y. Jia, and S. Liu, "A distributed Key Management Scheme Based on SGC-PKC for P2P Network," *Proc. Wireless Communications, Networking and Information Security (WCNIS), IEEE,* June 2010, pp.491-495.
[40] W. thomas, "Public Key Infrastructure Based on a Peer-to-Peer Network," *Proc. of System Sciences, IEEE,* January 2005, pp.1-10.

Felix Musau received BED (Honors) in Mathematics/Computer Science from Kenyatta University, Nairobi-Kenya, a Postgraduate Diploma in Computer Science and Technology, and MSC in Computer Science and Technology from Central South University, China. He is currently a Ph.D. candidate at the Trusted Computing Institute, Central South University, China. He is also a lecturer at the School of Engineering and Technology, Kenyatta University, Kenya. His current research interests include network and information security, trust management in peer to peer (P2P) e-commerce, and Internet of Things.

Guojun Wang received B.Sc. in Geophysics, M.Sc. in Computer Science, and Ph.D. in Computer Science, from Central South University, China. He is Head and Professor of Department of Computer Science at Central South University. He is also Director of Trusted Computing Institute at Central South University. He has been an Adjunct Professor at Temple University, USA; a Visiting Scholar at Florida Atlantic University, USA; a Visiting Researcher at the University of Aizu, Japan; and a Research Fellow at the Hong Kong Polytechnic University. His research interests include network and information security, Internet of Things, and cloud computing. He is a senior member of CCF, and a member of IEEE, ACM, and IEICE.



Shui Yu (M'05)received the B.Eng. and M.Eng. degrees from University of Electronic Science and Technology of China, Chengdu, China, in 1993 and 1999, respectively, and the Ph.D. degree from Deakin University, Victoria, Australia, in 2004. He is currently a Lecturer with the School of Information Technology, Deakin University, Victoria, Australia. His research interests include network security, networking theory, and mathematical modeling. He is a member of IEEE.

Muhammad Bashir Abdullahi received B.Tech (Honors) in Mathematics/Computer Science from Federal University of Technology, Minna-Nigeria andM.Sc.in Computer Science from Abubakar Tafawa Balewa University, Bauchi-Nigeria. He is currently a Ph.D. candidate at the Trusted Computing Institute, Central South University, P. R. China. His current research interests include trust, security and privacy issues in wireless sensor and ad hoc networks, Internet of things and network and information security.