Private and Flexible Urban Message Delivery

Shan Chang, *Member, IEEE*, Hongzi Zhu, *Member, IEEE*, Mianxiong Dong, *Member, IEEE*, Kaoru Ota, *Member, IEEE*, Xiaoqiang Liu, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—With the popularity of intelligent mobile devices, enormous amounts of urban information has been generated and demanded by the public. In response, ShanghaiGrid (SG) aims to provide abundant information services to the public. With a fixed schedule and urbanwide coverage, an appealing service in SG is to provide free message delivery service to the public using buses, which allows mobile device users to send messages to locations of interest via buses. The main challenge in realizing this service is to provide an efficient routing scheme with privacy preservation under a highly dynamic urban traffic condition. In this paper, we present the innovative scheme BusCast to tackle this problem. In BusCast, buses can pick up and forward personal messages to their destination locations in a store–carry–forward fashion. For each message, BusCast conservatively associates a routing graph rather than a fixed routing path with the message to adapt the dynamic of urban traffic. Meanwhile, the privacy information about the user and the message destination is concealed from both intermediate relay buses and outside adversaries. Both rigorous privacy analysis and extensive trace-driven simulations demonstrate the efficacy of the BusCast scheme.**

***Index Terms*—Anonymous communication, backward unlinkability, message delivery, traffic analysis attacks, vehicular networks.**

I. Introduction

With the prosperity of powerful intelligent mobile devices, e.g., tablets and smartphones, urban sensing information, such as photos of events and audio and video records, has been largely enriched. The ever-increasing demands for sharing such information from the public have become a serious challenge. In response to the challenge, in 2005, the Shanghai government established the ShanghaiGrid (SG) project, with the ambitious goal of building a metropolitan-scale information service system. Among others, one promising application in SG is to provide *message delivery service* in which mobile users can send messages to some locations of interest, such as homes, workplaces, and public agents. The goal of the application is threefold. First, it should guarantee anonymous data communication for users, which hides the privacy information stating who is communicating with whom and for what purpose from others. For example, *Alice* may take a picture of an event and would like to send it to the police station as evidence. Certainly, she will not send her ID information with the picture or want her identity information to be exposed by any means. Second, the end-to-end delivery may not have to be real time but should be short. Finally, the system should provide large service coverage to the public in terms of both geographical and temporal distributions.

To achieve the message delivery service, one possible solution is to use conventional cellular networks (e.g., GPRS and Third-Generation (3G)) or satellite techniques, which can provide very short delivery delay. However, the privacy of mobile users in terms of identities and their interested location information is not well protected from the network operators. In addition, it also causes tremendous communication cost for data transmission. Recently, vehicular networks [1]–[3] have emerged as the new landscape of mobile ad hoc networks, in which data communication is carried out in a *store–carry–forward* fashion. In SG, we consider to use buses (forming a bus network) for message delivery because of three major reasons. First, in urban settings, with the dense and wide distribution, commuting buses can reach very high coverage. For example in Shanghai, with a communication range of 600 m, the area covered by buses can be reached up to 90% of the downtown area. Second, the achievable end-to-end delay is convincing for most delay-tolerant applications with fixed bus routes and schedules. Finally, as buses are public vehicles, it is practical to provide such a service to the public without the concern of failures caused by selfish behavior.

To achieve efficient and anonymous message delivery with buses, however, is very challenging for three reasons. First, due to the dynamic surface traffic, buses may experience unexpected delays, which means that contacts between a pair of buses cannot be accurately predicted, even with the fixed bus schedules. In this case, simply using a predetermined shortest routing path calculated based on the static bus schedules is not feasible. Another naive solution can flood a message over the network, which can achieve the shortest delivery delay but lead to prohibitive network traffic. Second, because of the requirement of anonymous communication, all identification

Manuscript received November 17, 2014; revised June 20, 2015; accepted August 6, 2015. Date of publication August 11, 2015; date of current version July 14, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61300199, Grant 61202375, Grant 61472255, and Grant 61420106010; by the Fundamental Research Funds for the Central Universities under Grant 2232014D3-21; by the Innovation Program of Shanghai Municipal Education Commission under Grant 12ZZ060; by the Japan Society for the Promotion of Science) through Grants-in-Aid for Scientific Research (KAKENHI) under Grant 26730056 and Grant 15K15976; and through the A3 Foresight Program. The review of this paper was coordinated by Prof. Y. Fang.

S. Chang and X. Liu are with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: changshan@dhu.edu.cn; liuxq@dhu.edu.cn).

H. Zhu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200000, China (e-mail: hongzi@cs.sjtu.edu.cn).

M. Dong and K. Ota are with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-0071, Japan (e-mail: mx.dong@ieee.org; k.ota@ieee.org).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: xshen@bbcr.uwaterloo.ca).

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

Digital Object Identifier 10.1109/TVT.2015.2466651

information such as identities, locations, and routing paths must be removed from messages before being sent over open channels. Without the knowledge about the receiver, it is hard for an intermediate node to make an efficient routing decision. Finally, although identifying information can be well protected, it is much harder to defend against traffic analysis attacks, where adversaries observe the encrypted traffic flow to infer the relationship between messages. In consequence, messages can be traced forward to the destinations or backward to the sources [5]. One possible solution is to use onion routing [6], in which a message encrypted by the sender can be stripped off layer by layer by intermediate relay buses. Each relay bus can only get the information of its next hop and forward the message accordingly. This method needs the sender to determine the chain of all relay buses in advance. In the dynamic scenario of a bus network, however, it is hard to accurately predetermine an optimal routing path in terms of minimizing the delivery delay. Another alternative solution is to use universal reencryption [7], [8] in which each relay bus reencrypts a message without knowing the identity of the receiver. The solution needs to explicitly provide the identity of the receiver to relay buses, which would disclose the privacy of the receiver. As a result, there is no successful solution, to the best of our knowledge, to provisioning efficient private message delivery in bus networks.

In this paper, we propose an innovative message delivery scheme in bus networks called *BusCast*, which provides a set of routing mechanisms flexible to the uncertainty of bus contacts caused by dynamic surface traffic while provisioning anonymous communication for users. BusCast elegantly integrates three key techniques. First, BusCast users can plan a routing graph for a message, which is made up of a set of relay rules indicating how the packet would transfer between bus routes. Second, we design an anonymous routing structure (ARS) to indicate routing information for intermediate relay buses by embedding the routing graph in the message. With the ARS, relay buses can only recover their own routing instructions about which bus routes are the next hop presented in the routing paths. Finally, each relay bus regenerates the ARS, and the confidential messages of users are also reencrypted on each relay bus using a universal reencryption scheme, which requires no public key of message receivers. Combining the two together eliminates the linkability between incoming and outgoing packets, which can defend against traffic analysis attacks. The strong point of BusCast design is that even if the secrets of one or more bus routes are exposed, anonymous communication can still be achieved. Thorough privacy analysis shows that the BusCast design can protect user privacy well. We also verify the routing performance of BusCast though extensive trace-driven simulations that involve 199 bus routes in Shanghai city.

We highlight our main contributions in this paper as follows.

- We have considered the dynamic of suffice traffic in realizing the message delivery service and the design an ARS, which conceals identification information from other buses and outside adversaries and provides plenty of flexibility in making the routing decision.
- We allow intermediate buses embedded in an ARS to regenerate the ARS without the needing know the routing graph, which makes buses act as both message routers and mix proxies of a mix net [4] and, therefore, can defend against traffic analysis attacks and guarantee anonymous communications.
- We have conducted both privacy and performance analysis, as well as extensive trace-driven simulations, to demonstrate the efficacy of the BusCast design.

The remainder of this paper is organized as follows. Section II introduces related work. In Section III, we characterize the unique features of bus networks. Section IV describes the system and attack models in bus networks and presents the design goals. Section V present the overview of BusCast. We elaborate the technique of flexible routing with routing graph in Section VI. In Section VII, the details of privacy-preserving packet forwarding using ARS are described. Section VIII presents the privacy and performance analysis of BusCast. Several design issues that may be encountered in practice are discussed in Section IX. In Section X, we conduct trace-driven simulations to evaluate the performance of BusCast and present the results. Finally, we conclude and outline the directions for future work in Section XI.

II. Related Work

Since the concept of mix net was first introduced by Chaum for anonymous communications on the Internet [4], many studies have followed Chaum's approach, such as Web-MIXes [9], Tarzan [10], Mixminion [11], and AOS [12]. A mix node is a proxy that batched modifies input messages and outputs them in a random order, called mixing. In this way, it is hardly to correlate a comes in message with a goes out message, which can be leveraged to defend against the traffic analysis attacks.

Several mixing schemes are designed for providing anonymous routing in mobile networks, such as ANODR [13], SDAR [14], and AnonDSR [15]. These schemes share two common features. First, all these schemes demand a route discovery phase before forwarding a packet, which enables the sender to discover and establishes a secure routing path via a number of intermediate wireless nodes to the receiver. Second, all these schemes use the layer-by-layer encryption/decryptions. The significant feature of these schemes is that the layers should be peeled in sequence, and at each time, only one layer ca be peeled. However, the strictly defined routing path lacks the flexibility to adapt rapid changes of mobility of vehicular networks. Fan *et al.* [16] proposed a network coding scheme combining with homomorphic encryption functions to protect the source anonymity from traffic analysis and flow tracing attacks in multihop wireless networks. They considered a multicast network. Intermediate nodes buffer the received packets until all the packets belonging to the same session are available and perform random linear coding on these packets. However, the scheme cannot work well in vehicular networks since the opportunistic routing makes it hard to collect all packets belonging to the same session at the same intermediate.

There are two works closely related to this paper. First, the work of Jansen and Beverly introduced a threshold pivot scheme [17] to address the anonymity issues in delay-tolerant networks. In the scheme, nodes are divided into several groups. Senders generate a one-time secret key to encrypt each message

Fig. 1. Distribution of bus lines within the downtown area of Shanghai City, with 199 bus lines involved.

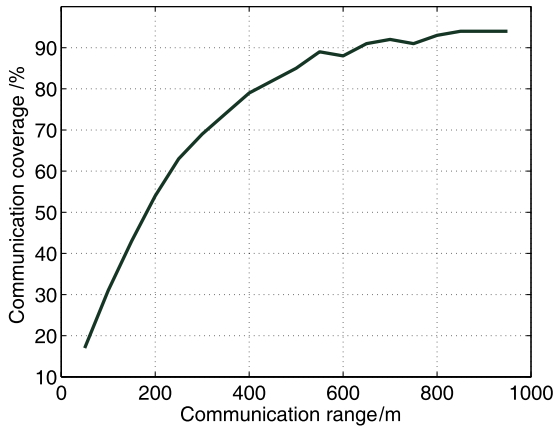


Fig. 2. Bus network coverage under different communication ranges.

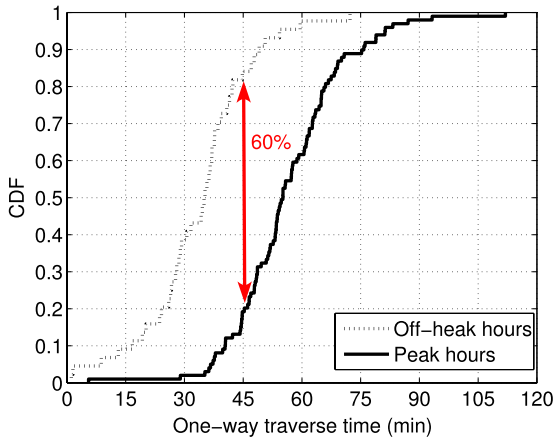


Fig. 3. Comparison of one-way traverse time of buses between off-peak and peak hours.

and the identities of receivers. Senders use a secret sharing scheme to divide the secret into pieces, and each piece is encrypted using a group key. Each relay node decrypts a piece of secret using the group key it holds. Node decrypting the last piece of secret can recover the original secret key. Then, the message and identity of receiver can be revealed and be routed to receiver. However, this scheme leaks identities of receivers to a part of untrusted intermediate nodes. Lu *et al.* put forward a social-based privacy-preserving packet-forwarding protocol in vehicular ad hoc networks [8]. They deployed roadside units (RSUs) at high social degree intersections to assist in packet forwarding between vehicles by temporarily storing packet and to carry out reencryption on the packet to construct mix network. However, locations of receivers are provided to all relay RSUs and vehicles. Furthermore, to protect the privacy, each packet should be at least temporarily stored on and reencrypted by an RSU.

III. Characteristics of a Bus Network

Understanding the properties of the network composed of urban buses is essential to the performance of message delivery. Here, we study the key features of a bus network based on a real global positioning system (GPS) trace collected from 2358 buses on 199 bus lines between February 19 and March 5, 2007. It covers the downtown area of Shanghai City, which is about 120 km^2.

We first illustrate the geographical distribution of all bus lines in Fig. 1, where the red lines represent the aggregated itineraries of all bus lines. It can be seen that bus lines show a dense and relatively uniform distribution throughout the whole region. We refer to the coverage of the bus network as the area that messages can be delivered by a bus of the network, using short-range wireless communication (e.g., Dedicated Short Range Communications: DSRC, i.e., 802.11p).

Fig. 2 shows the coverage as the function of the wireless communication range. It can be seen that, when the communication range is above 600 m, the coverage ratio can reach over 90% of the total area. In addition, it can also be seen that bus lines are well interconnected with different bus lines, sharing part of their routes in common. Both properties indicate that the bus network is ideal for message delivery.

Ideally, buses travel between stops on their routes on time no matter how the surface traffic changes during the day. In this case, a simple yet effective way to deliver a message is to forward the message along the shortest path calculated based on the fixed bus routes and regular schedules. In reality, however, the mobility of buses varies dramatically at different time, particularly in metropolises such as Shanghai. For example, Fig. 3 shows the cumulative distribution function of the traverse time for a bus to travel from one terminal to the other on its route over all buses during the peak and off-peak hours in a day, respectively. It can be seen that 80% buses can finish one-way traversing within 45 min during off-peak hours, but the ratio drops dramatically to about 20% during peak hours. The huge variation makes predetermining the shortest routing path very hard, if not impossible.

IV. Models and Design Goals

A. System Model

In BusCast, there are two components: buses and trust authority (TA).

- *Buses* are equipped with onboard units (OBUs), which typically consist of a CPU, a large storage device, a GPS

module, and wireless communication modules. Moving buses can talk with other buses via short-range wireless communication (e.g., DSRC) and with the TA via long-range wireless communication (e.g., GPRS, 3G).
- *TA* is a trustworthy authority, which can communicate with buses all the time. TA generates common secret information for buses on the same route.

We assume that buses carry out their functionality properly but may cause secret information generated by TA leaked under intrusion of adversaries. We assume each bus has an intrusion detection system (IDS). If an intrusion is detected, it reports to the TA to take corresponding security responses. We also assume that information on routes is available to the public (e.g., from the website).

B. Attack Model

We characterize adversaries from four perspectives. First, adversaries can mount both passive and active attacks, which implies not only eavesdropping but also packets injection and modification on the wireless channel. Second, adversaries can have the global view of the whole network traffic by eavesdropping on the open channel. Third, adversaries behave rationally, which means they launch attacks to gain benefits. The goal of adversaries is to jeopardize use's privacy. In particular, we consider privacy jeopardizing attacks in two aspects.

- *Confidentiality violation*: Attackers eavesdrop on the shared medium to catch other people's communications and recover the content of packets to obtain the confidential information.
- *Anonymity violation*: Even if the content of packets is protected by encryptions, attackers can also obtain sensitive information related to identities, e.g., locations of victims, by launching traffic analysis or packet marker attacks. In traffic analysis attacks, adversaries intercept and examine encrypted messages to deduce information from patterns in communication. In packet marker attacks, adversaries insert some distinguishable markers into packets to track them.

Finally, although intrusions of attackers can be detected by IDSs, we do not assume that the damage results (secret exposure) can be avoided perfectly. Adversaries can make use of the exposed secrets by trying to recover other sensitive information. Additionally, we assume that the invaded routes should be a small fraction of all the bus routes.

C. Design Goals

The BusCast design should meet both privacy-related and routing performance-related requirements.

1) Privacy Requirements: We aim to enable anonymous message delivering. More specifically, the following properties should be guaranteed.

- *Unlinkability* between users and destinations should be guaranteed, which means that it is not possible to trace who communicates with whom.
- *Backward unlinkability* between users and destinations should be guaranteed, which means that, even after some buses are invaded and consequently get secret exposed, past communications remain untraceable.
- *Confidentiality* of the messages should be protected. Sensitive information shared among senders and receivers will never be disclosed to others.

2) Routing Performance Requirements: Since the number of potential users may be very large, it is of great importance to consider the scalability of the system. The BusCast design should minimize the message delivery delay and the corresponding network cost that occurs.

V. Overview of BusCast Design

To tackle the challenges in realizing message delivery service, BusCast elegantly integrates three techniques: *flexible routing with routing graph*, *constructing ARS*, and *regenerating ARS, and reencrypting user data (to form mix net)*.

Specifically, BusCast uses a routing graph instead of one single shortest path to forward a message, in which each edge indicates how the packet would transfer between bus routes. By elaborately constructing the routing graph for the message, plenty of flexibility can be achieved to adapt the uncertainty of bus mobility caused by dynamic suffice traffic. Given a routing graph of the message, we construct and associate an ARS with the message to indicate routing information for intermediate relay buses by associating the routing graph with the message. With the ARS, relay buses can only recover their own routing instructions about which bus routes are the next hop presented in the routing paths. To break the linkability between incoming and outgoing packets, each relay bus regenerates the ARS without the need to know the routing graph. Furthermore, the confidential messages of users are also reencrypted on each relay bus using a universal reencryption scheme, which requires no public key of message receivers. Without linkability during the routing process, BusCast can defend against traffic analysis attacks.

VI. Flexible Routing With Routing Graph

A. Contact Graph of Buses

Since buses on the same route share the same itinerary, we consider message delivery problem at the level of bus routes. Two bus routes are referred to as *neighbors* if they share partial itinerary or have intersections.

Suppose that a mobile device user *Bob* wants to send a message m to his friend *Alice* using bus networks, and *Bob* knows her location $\mathcal{L}$. *Bob* forwards m to a nearby passing bus; then, m will be relayed between buses and finally reach a bus route passing through $\mathcal{L}$. Fig. 4 shows an example of a message delivery using bus networks. The different colors of solid lines denote different route paths, and the number in each circle represents the route ID.

We use a *contact graph* $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{W})$ to represent contacts between bus routes. A contact graph consists of a set of vertices $\mathbb{V}$, a set of edges $\mathbb{E}$, and a set of weights $\mathbb{W}$, each of which is assigned to an edge of the graph. Since route paths are

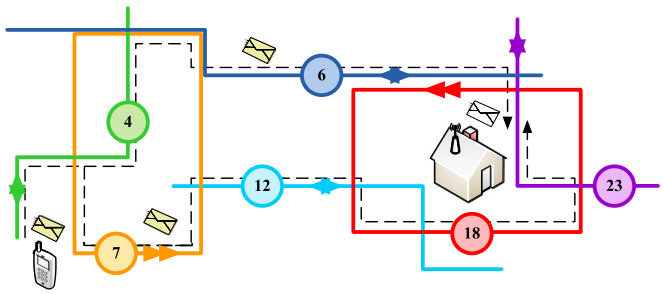


Fig. 4. Example of the packet-forwarding procedure using the bus network.

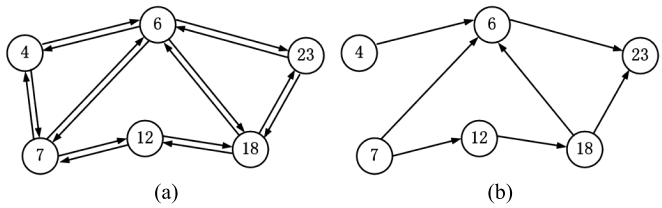


Fig. 5. Example (the bus network in Fig. 4) of the contact graph and routing graph. (a) Contact graph $\mathbb{G}$. (b) Example of routing graph $\mathbb{G}'$.

public information, it is convenient to determine $\mathbb{V}$ and $\mathbb{E}$ using neighbor relationship between routes.

- Each vertex $v_i \in \mathbb{V}$ denotes a bus route (line).
- If two bus lines $v_i, v_j \in \mathbb{V}$ are neighbors, there are two edges $\hat{e}(v_i, v_j)$ and $\hat{e}(v_j, v_i) \in \mathbb{E}$ between v_i and v_j (edges between two vertices come in pairs since neighborship is bidirectional).

Fig. 5(a) shows the contact graph $\mathbb{G}$ of the bus network in Fig. 4. According to the static and dynamic features of buses, $w_{i,j} \in \mathbb{W}$ can be set in two ways. Assume that the length of bus line v_i is l_i; there is an edge $\hat{e}(v_i, v_j) \in \mathbb{E}$, and v_i and v_j share x intersections and several road segments. The total length of share road segments is y. The communication range between buses is r. According to the given static geographical feature of v_i and v_j, $w_{i,j} = 1/((r/l_i) \cdot x + (y/l_i))$, where $(1/((r/l_i) \cdot x + (y/l_i)))$ is an approximation of overlap ratio of v_i and v_j, and it is proportional to the contact opportunities between v_i and v_j. Hence $w_{i,j}$ is proportional to their intercontact time. A more realistic (dynamic) way to determine $w_{i,j}$ is using the intercontact time of routes directly, which denotes the average time elapsed between two successive contacts of a certain bus on v_i and any one bus on route v_j.

B. Extracting Routing Graph

To deliver m to *Alice* effectively, *Bob* should indicate (implicit or explicit) location (or identity) information of *Alice* to relay buses. Basically, there are two kinds of routing strategies. First, *Bob* indicates $\mathcal{L}$ explicitly in the packet. However, it is prohibited for privacy concerns. Second, *Bob* plans a routing path using $\mathbb{G}$ of the bus network in advance (called deterministic routing) and uses particular privacy-preserving techniques, such as onion routing, giving each relay bus knowledge of only a small part of the fix routing path. However, deterministic routing, such as computing a shortest path, usually does not work well due to the large variation of contact time and locations between buses, which can tremendously deviate the values predicted using bus schedules.

Consequently, we introduce flexibility in deterministic routing. First, senders designate a number of relay instructions between neighbor bus lines for each packet rather than a fixed routing path, according to $\mathbb{G}$ and performance requirements. Relay instructions form a *routing graph* $\mathbb{G}' = (\mathbb{V}', \mathbb{E}', \mathbb{W}')$, which is an induced subgraph of $\mathbb{G}$, i.e., $\mathbb{V}' \subseteq \mathbb{V}$, $\mathbb{E}' \subseteq \mathbb{E}$, and $\mathbb{W}' \subseteq \mathbb{W}$. Directed edges $\hat{e}(v_i, v_j) \in \mathbb{E}'$ denote a relay rule. $\mathbb{G}'$ contains at least one *source vertex* $v_s : \forall\, v_i \in \mathbb{V}', \nexists \hat{e}(v_i, v_s) \in \mathbb{E}'$ and one *destination vertex* $v_d : \forall\, v_i \in \mathbb{V}', \nexists \hat{e}(v_d, v_i) \in \mathbb{E}'$. For the example in Fig. 4, a user is located near bus lines 4 and 7. The user wants to send a packet to a place located near bus line 23. Fig. 5(b) gives an example of $\mathbb{G}'$. A simple method to extract $\mathbb{G}'$ is to pick the k-shortest paths from v_s to v_d on $\mathbb{G}$ (removing duplicate edges). Sophisticated methods can be used if more information is available, e.g., the schedule of buses.

During message delivering, senders deliver a packet to a passing-by bus on route v_s; then, the bus carries the packet until encountering v_i that $\hat{e}(v_s, v_i) \in \mathbb{E}'$ and forwards the packet to v_i. In the same way, the packet is relayed by buses in a route-by-route manner according to $\mathbb{G}'$ until reaching v_d. Then, the packet is carried by v_d and broadcast in the vicinity of destination $\mathcal{L}$.

VII. Privacy-Preserving Packet Forwarding Using Anonymous Routing Structure

A. System Initialization

Given a contact graph $\mathbb{G}$, each edge $\hat{e}(v_i, v_j) \in \mathbb{E}$ is mapped to a routing instruction, referring to a *relay indicator* (RI, denoted I_{rly}), to indicate a one-hop relay from v_i to v_j. We also define *broadcast indicators* (BI, denoted I_{bst}), which are used to indicate broadcasting areas for the buses.

Each indicator (BI or RI) has two parts: a *public indicator* I^p and a *secret indicator* I^s. Public indicators are known to all entities and are used for constructing ARSs. Secret indicators are kept by certain routes secretly for verifying ARSs. It is similar with public/private key pairs; however, indicators serve to build ARS rather than encrypt messages.

TA is responsible for constructing all the indicators in the system. In addition, TA generates the public parameters and master key of the system.

1) Generating Public Indicators: For each route R_i, TA assigns a set of public RIs for R_i corresponding to edges $\hat{e}(v_i, v_j) \in \mathbb{E}$ in $\mathbb{G}$. In other words, each RI relates to a neighbor of R_i. For its neighbor R_j, the corresponding public RI is $\mathbf{I^P_{rly(R_i,R_j)}} = \mathbf{R_i} \| \mathbf{R_j}$, which **indicates one-hop relay from $\mathbf{R_i}$ to $\mathbf{R_j}$**. The symbol $\|$ represents concatenation between strings.

For example in Fig. 4, R_7 has three neighbors that are R_4, R_6, and R_{12}. TA assigns three public RIs $I^p_{\text{rly}(R_7,R_4)}$, $I^p_{\text{rly}(R_7,R_6)}$, and $I^p_{\text{rly}(R_7,R_{12})}$ for R_7. It should be noted that neighbors do not share RIs. R_4 and R_7 hold $I^p_{\text{rly}(R_7,R_4)} = R_7 \| R_4$ and $I^p_{\text{rly}(R_4,R_7)} = R_4 \| R_7$ separately.

TA also assigns a set of BIs for R_i representing certain areas where it should broadcast packets. We divide the urban area

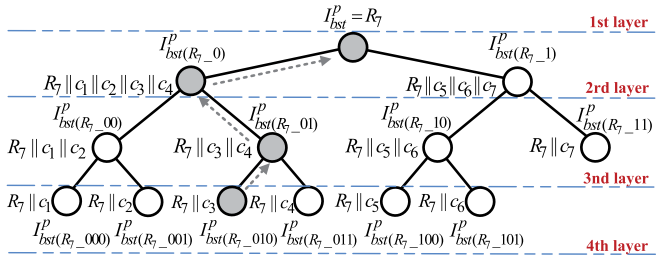


Fig. 6. Public BI tree of route R_7 that covers seven cells $\{c_1, c_2, \ldots, c_7\}$.

into small cells according to the communication range of buses. Each cell has a unique identity. Cells within the coverage area of R_i are organized in sequence of locations from its departure to the terminal station. Assume that n cells are in the coverage area of R_i, which are $\{c_1, c_2, \ldots, c_n\}$. TA executes the following operations to construct a binary tree of public BIs of R_i so that a BI can be targeted efficiently. The height of the BI tree is $\lceil \log_2 n \rceil + 1$:

- The root node of the tree is $I^p_{\text{bst}(R_i)} = R_i$, which is used to indicate whether R_i is one of the last relay routes.
- Split the original set of cells into $\{c_1, c_2, \ldots, c_{\lceil n/2 \rceil}\}$ and $\{c_{\lceil n/2 \rceil+1}, c_{\lceil n/2 \rceil+2}, \ldots, c_n\}$. The first-layer public BIs are $R_i \| c_1 \| \cdots \| c_{\lceil n/2 \rceil}$ and $R_i \| c_{\lceil n/2 \rceil+1} \| \cdots \| c_n$, which are denoted $I^p_{\text{bst}(R_i\_0)}$ and $I^p_{\text{bst}(R_i\_1)}$.
- Execute the split recursively on two resulting sets, and obtain the corresponding public BIs until only one cell is left in each set.

For example, in Fig. 4, assume R_7 covers seven cells, which are $\{c_1, c_2, \ldots, c_7\}$. The four-layer public BI tree of R_7 is shown in Fig. 6. $R_i \| c_j \| \cdots \| c_k$ indicates that the destination cell belongs to the set of $\{c_j, \ldots, c_k\}$.

2) Generating Secret Indicators: For a given public indicator $I^p \in \{0,1\}^*$ (either I^p_{rly} or I^p_{bst}), TA computes $Q_I = H_1(I^p) \in \mathbb{G}_1^*$ and sets the corresponding secret indicator as $I^s = s(Q_I + P_1)$. For example, given $I^p_{\text{rly}(R_7,R_6)} = R_7 \| R_6$, TA generates $I^s_{\text{rly}(R_7,R_6)} = s(H_1(R_7 \| R_6) + P_1)$; for $I^p_{\text{bst}(R_7\_00)} = R_7 \| c_1 \| c_2$, corresponding secret BI $I^s_{\text{bst}(R_7\_00)}$ is $s(H_1(R_7 \| c_1 \| c_2) + P_1)$. TA distributes secret indicators to the bus routes to which they belong via a secure channel (e.g., secret indicators are signed and encrypted using a signcryption scheme [18], and transmitted using a 3G network).

3) Generating System Parameters: Given a security parameter $n \in \mathbb{Z}^+$, TA runs the following algorithms to generate the public parameters and master key in the system.

a) Generate a large prime q, an additive cyclic group $\mathbb{G}_1$ and a multiplicative cyclic group $\mathbb{G}_2$ of order q, and a *bilinear mapping* $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ [19]. Choose a random generator $P \in \mathbb{G}_1$.
b) Pick $s \xleftarrow{R} \mathbb{Z}_q^*$ as a master key, and set $P_{\text{pub}} = sP$.
c) Pick $P_1 \xleftarrow{R} \mathbb{G}_1$.
d) Choose cryptographic hash functions:

$$H : \{0,1\}^* \to \mathbb{Z}_q^*, H_1 : \{0,1\}^* \to \mathbb{G}_1^*$$
$$H_2 : \mathbb{G}_2 \to \{0,1\}^\omega, H_3 : \{0,1\}^\omega \times \{0,1\}^* \to \mathbb{Z}_q^*.$$

e) Set $\mathbf{Par}_A = \{q, \mathbb{G}_1, \mathbb{G}_2, e, P, P_1, P_{\text{pub}}, H, H_1, H_2, H_3\}$.
f) Pick another large prime p that states that discrete logarithm problems are difficult on $\mathbb{Z}_p^*$, and $g \in \mathbb{Z}_p^*$, which is a primitive root of $\mathbb{Z}_p^*$. Set $\mathbf{Par}_E = \{p, g\}$.

Then, the TA publishes all public indicators $\mathbf{Par}_A$ and $\mathbf{Par}_E$ while keeping the master key s secret.

B. Operations Conducted by Message Senders

Recall the example of *Bob* and *Alice* in Section VI. After extracting the routing graph $\mathbb{G}'$ of m according to $\mathbb{G}$, current location and $\mathcal{L}$, *Bob* encrypts m and constructs ARS for m.

1) Encrypting User Data m: *Bob* uses the universal reencryption scheme proposed by Golle *et al.* [7] to encrypt m. The scheme has the feature that intermediate nodes can reencrypt m without knowing the identity or public key of the receiver. Specifically, *Bob* uses *Alice*'s public key $pk_A = g^{sk_A} (sk_A \in \mathbb{Z}_p)$ to encrypt m, so that only *Alice* who holds the secret key sk_A can recover m. The ciphertext $\mathcal{M}$ has two parts. The second part is used for future reencryptions. The encryption proceeds as follows.

a) Pick a pair of random encryption factors $(\tau_1, \tau_2) \in \mathbb{Z}_p^2$.
b) Compute $\mathcal{M} = \{(m \cdot pk_A^{\tau_1}, g^{\tau_1}); (pk_A^{\tau_2}, g^{\tau_2})\}$.

2) Constructing ARS: Assume the routing graph $\mathbb{G}'$ includes μ routes $\{R_1, R_2, \ldots, R_\mu\}$; R_μ is the last relay route, which covers n cells $\{c_1, c_2, \ldots, c_n\}$; and $\mathcal{L}$ is located in cell $c_\rho (1 \le \rho \le n)$. Then, *Bob* imbeds RI and BI into ARS, according to edges of $\mathbb{G}'$ and c_ρ. Specifically, *Bob* carries out the following steps to pick public RIs and BIs.

a) Choose public RIs: If $e(v_{R_i}, v_{R_j}) \in \mathbb{G}', 1 \le i, j < \mu$, then pick $I^p_{\text{rly}(R_i,R_j)} = R_i \| R_j$.
b) Choose public BIs:

- Pick $I^p_{\text{bst}(R_\mu \_ \{0,1\}^{\lceil \log_2 n \rceil})} = R_\mu \| c_\rho$.
- Pick all public BIs on the path from $R_\mu \| c_\rho$ to the root of BI tree, i.e.,

$$\left\{ I^p_{\text{bst}\left(R_\mu \_ \{0,1\}^{\lceil \log_2 n \rceil - 1}\right)}, \ldots, I^p_{\text{bst}(R_\mu \_ \{0,1\})}, I^p_{\text{bst}(R_\mu)} \right\}$$

For the example in Fig. 6, if $\mathcal{L}$ is in the cell c_3, then $I^p_{\text{bst}(R_7\_010)}$, $I^p_{\text{bst}(R_7\_01)}$, $I^p_{\text{bst}(R_7\_0)}$, and $I^p_{\text{bst}(R_7)}$ (gray circles) will be selected as public BIs.

Assume *Bob* has chosen t public indicators $I^p_i, (1 \le i \le t)$ (both RIs and BIs); he generates the ARS as follows:

a) For each I^p_i, compute $x_i = H(I^p_i)$ and $Q_i = H_1(I^p_i)$.
b) Compute

$$\ell_i(x) = \prod_{1 \le j \ne i \le t} \frac{x - x_j}{x_i - x_j} = a_{i,1} + a_{i,2}x + \cdots + a_{i,t}x^{t-1}$$

where $a_{i,1}, a_{i,2}, \ldots, a_{i,t} \in \mathbb{Z}_q$. Then

$$\ell_i(x_j) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \ne j. \end{cases}$$

TABLE I
ALGORITHM 1

Algorithm1: Verify$(I^p, I^s, ARS, \mathcal{M})$

/*$ARS = \langle \mathbb{B}, \alpha P, \beta P_{pub}, \sigma \oplus H_2(e(P_{pub}, P_1)^{\alpha})\rangle$ */
$x = H(I^p)$;
$\theta = B_1 + xB_2 + ... + (x^{t-1} \mathbf{mod} q)B_t$;
$\bar{\sigma} = \sigma \oplus H_2\left(e\left(P_{pub}, P_1\right)^{\alpha}\right) \oplus H_2\left(\frac{e(\alpha P, I^s)}{e\left(\beta P_{pub}, \theta\right)}\right)$;
$\bar{\alpha} = H_3(\bar{\sigma}, \mathcal{M})$;
If $\bar{\alpha}P = \alpha P$, return **TRUE**; /* I^p was imbedded in ARS and $\bar{\alpha} = \alpha$*/
Else return **FALSE**.

c) Set $\mathbf{a}_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,t}]^T$. Compute the routing information

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_t \end{bmatrix} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_t] \times [Q_1, Q_2, \ldots, Q_t]^T = \begin{bmatrix} a_{1,1}Q_1 + a_{2,1}Q_2 + \cdots + a_{t,1}Q_t \\ a_{1,2}Q_1 + a_{2,2}Q_2 + \cdots + a_{t,2}Q_t \\ \vdots \\ a_{1,t}Q_1 + a_{2,t}Q_2 + \cdots + a_{t,t}Q_t \end{bmatrix}$$

d) Pick a string $\sigma \overset{R}{\leftarrow} \{0,1\}^{\omega}$, and set $\alpha = H_3(\sigma, \mathcal{M})$.
e) Pick an integer $\beta \overset{R}{\leftarrow} \mathbb{Z}_q^*$, and set $y = (\beta^{-1}\alpha) \mathbf{mod} q$.
f) Compute $y \cdot [A_1, A_2, \ldots, A_t] = [B_1, B_2, \ldots, B_t] = \mathbb{B}$.
g) Set ARS $= \langle \mathbb{B}, \alpha P, \beta P_{\text{pub}}, \sigma \oplus H_2(e(P_{\text{pub}}, P_1)^{\alpha})\rangle$.

Then, *Bob* sets message $C = \langle \text{ARS}, \mathcal{M}\rangle$ and forwards C to one source route v_s in $\mathbb{G}'$ directly.

C. Operations Conducted by Buses

When a bus on R_i receives message C, it verifies whether R_i was selected as a relay route. If so, the bus reconstructs C, and relays the updated C according to the indicators.

1) Verifying ARS: The bus on R_i uses its public/secret indicator pairs (I^p, I^s) to verify which indicators were imbedded in the ARS. It checks the ARS as follows.

a) Verify if the root of R_i's public BI tree $I^p_{\text{bst}(R_i)}$ was imbedded in ARS, which means R_i is the last relay route of C. If so, go to step 2; otherwise, go to step 3.
b) Verify if its second-layer BIs $\left\{I^p_{\text{bst}(R_i\_0)}, I^p_{\text{bst}(R_i\_1)}\right\}$ were imbedded in ARS. One of them is included in ARS means that C should be broadcast within the corresponding region. If one of them was in ARS, R_i further verifies if any of its two children BIs was imbedded in ARS, and so on, until reaching to the highest layer BIs. The highest layer BI imbedded in ARS indicates the final broadcasting cell.
c) Verify RIs of R_i to see if some of them were imbedded in ARS, which means R_i is one of intermediate relay routes, and C should be sent to one of the routes included in the corresponding RIs.

Given an indicator pair (I^p, I^s), the verification can be done using **Algorithm 1** (see Table I).

If all verifications return false, it means that R_i is not in $\mathbb{G}'$. Hence, R_i simply ignores it. Otherwise, R_i should be either an intermediate or one of the last relay route.

2) Regeneration of ARS and $\mathcal{M}$: If R_i is an intermediate relay route, the bus on R_i regenerates the ARS and $\mathcal{M}$ to build the mix net, thus defending against traffic analysis attacks. Then, R_i prepares $C = \langle \text{ARS}, \mathcal{M}\rangle$ for the next relay. If R_i is the last relay route, (i.e., R_μ) the bus simply discards the ARS and reencrypts $\mathcal{M}$. Then, R_i broadcasts $\mathcal{M}$ in cell c_ρ calculated before. The reencryption of $\mathcal{M}$ proceeds as follows.

a) Choose a random reencryption factor $(\tau_1', \tau_2') \in \mathbb{Z}_p^2$.
b) Compute

$$\mathcal{M}' = \left\{\left(m \cdot pk_A^{\tau_1}(pk_A^{\tau_2})^{\tau_1}, g^{\tau_1}(g^{\tau_2})^{\tau_1'}\right); \left((pk_A^{\tau_2})^{\tau_2'}, (g^{\tau_2})^{\tau_2'}\right)\right\} = \{(\lambda_0, \rho_0); (\lambda_1, \rho_1)\}.$$

The regeneration of the ARS proceeds as follows.

a) Pick a string $\sigma' \overset{R}{\leftarrow} \{0,1\}^{\omega}$, and set $\alpha' = H_3(\sigma', \mathcal{M}')$.
b) Pick an integer $\beta' \overset{R}{\leftarrow} \mathbb{Z}_q^*$.
c) Note RID$_i$ can obtain the value of α from **Algorithm 1**. Set $y' = ((\beta')^{-1}\alpha^{-1}\alpha') \mathbf{mod} q$.
d) Compute

$$\begin{aligned} \mathbb{B}' &= y' \cdot \mathbb{B} = y'y \cdot [A_1, A_2, \ldots, A_t] \\ &= \left((\beta'\beta)^{-1}\alpha'\right) \mathbf{mod} q \cdot [A_1, A_2, \ldots, A_t] \\ &= [B_1', B_2', \ldots, B_t']. \end{aligned}$$

e) ARS$' = \left\langle \mathbb{B}', \alpha' P, \beta\beta' P_{\text{pub}}, \sigma' \oplus H_2(e(P_{\text{pub}}, P_1)^{\alpha'})\right\rangle$.

D. Operations conducted by Receivers

When *Alice* receives $\mathcal{M}'$, she decrypts $\mathcal{M}'$ using her private key sk_A as follows.

a) Verify if $\lambda_1/\rho_1^{sk_A} = (pk_A^{\tau_2})^{\tau_2'}/((g^{\tau_2})^{\tau_2'})^{sk_A} = 1$. If so, decryption is successful; then, execute step 2. Otherwise, decryption fails; drop the packet.
b) Compute $m = \lambda_0/\rho_0^{sk_A}$.

Fig. 7 shows the whole procedure of data forwarding using ARS, including the encryption of user data, construction of ARS, verification of ARS, regeneration of ARS and ciphertext $\mathcal{M}$, and decryption of user data.

VIII. ANALYSIS

A. Privacy Analysis

We first analyze abilities of different entities on understanding packets, which facilitates privacy analysis of the proposed scheme. First, for buses in the $\mathbb{G}'$ of a message $C(C')$, after receiving the message, they can verify that some of their indicators are imbedded in ARS. For the intermediate and the last relay routes in $\mathbb{G}'$, they can get one-hop routing instructions and broadcasting region of the packet, respectively. Each route in $\mathbb{G}'$ can only reveal indicators of their own; no one has the knowledge of a whole routing graph. Second, for buses that are

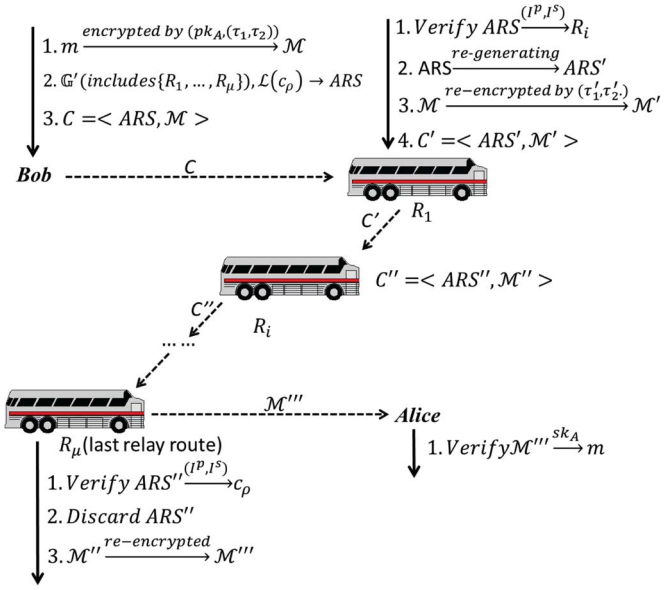


Fig. 7. Data forwarding (*Bob* sends m to *Alice*) using ARS.

not in $\mathbb{G}'$, after receiving $C(C')$, they verify the ARS in $C(C')$, and all verification fails. Hence, they only know that they are not in $\mathbb{G}'$. Third, because of the hop-by-hop reconstruction of C, neither inside nor outside observers can link C to C'. No one has a global view on the routing path of C. Although adversaries have a global view of traffic in bus networks, they can hardly induce the travelling path of single packet. Hence, the mix net is constructed from both the viewpoint of outside and inside observers. Finally, m is encrypted by *Alice*'s public key; none of them can decrypt $\mathcal{M}(\mathcal{M}')$ except *Alice*. Since the confidentiality can be achieved obviously, we focus on analyzing the unlinkability and backward unlinkability of the proposed method.

1) Unlinkability: Adversaries cannot launch traffic analysis attacks successfully by monitoring communication channel for two reasons. First, buses serving as mix proxies reconstruct all communications in each hop. Content relevance between incoming and outgoing packets on buses is wiped off. Hence, adversaries cannot link different packets by their contents. Second, buses pick and forward packets in a store–carry–forward fashion. Buses receive packets continuously; thus, a number of packets are stored waiting for transmission. Once a relay opportunity arises or a bus approaches a broadcasting region, packets under the same indicator stored on the bus will be relayed all together. This way, the spatial and temporal correlations between incoming and outgoing packets on buses are eliminated.

Adversaries cannot launch packet marker attacks to track packets. In such attacks, an adversary eavesdrops packet C and replaces $\mathcal{M}$ with an encrypted marker $\mathcal{N}$ that can be decrypted by it. However, Relay buses can verify that the ARS is not to be generated for $\mathcal{M}$ (since $(\bar{\alpha}' = H_3(\bar{\alpha}, \mathcal{N})$; however, $\bar{\alpha}' P \neq \alpha P$). Then, $\mathcal{N}$ will be dropped.

2) Backward Unlinkability: If an adversary invades a bus in $\mathbb{G}'$ and obtains packet C recorded by it, by verifying ARS, it can only recover one-hop routing information. Moreover, any relay bus reencrypting $\mathcal{M}$ does not know the identity of *Alice*, which implies that the adversary cannot get any information of *Alice* by revisiting $\mathcal{M}$. Hence, backward unlinkability can be achieved.

TABLE II
NOTATIONS

Notaton	Meaning
t	The number of public indicators imbedded on ARS
e	Bilinear map
ϕ	Bit length of elements in $\mathbb{G}_1$
ω	Bit length of strings outputted by $H_2(\cdot)$
GA	Addition over $\mathbb{G}_1$
GM	Multiplication over $\mathbb{G}_1$
GE	Exponentiation over $\mathbb{G}_2$
EX	Exponentiation over $\mathbb{Z}_q^*$
MUL	Multiplication over $\mathbb{Z}_q^*$

B. Performance Analysis of Privacy-Preserving Mechanisms

The proposed scheme contains operations related to ARS and m. The performance of the reencryption scheme on m has been analyzed in [7]. Hence, we focus on the performance of ARS operations. Table II summarizes the notations used.

Since $\mathbb{G}$ and cells are static for a given bus network, all RIs and BIs over the network are determined. Hence, senders can conduct one-time computations on $x_i = H(I_i^p)$ and $Q_i = H_1(I_i^p)$ for all indicators I_i^p in advance. (x_i, Q_i) can be used on all future ARS constructions. For this reason, the generation of (x_i, Q_i) was not counted in the computation cost. Assume t public indicators are imbedded in an ARS. Computational complexity and storage complexity of the ARS are summarized in Table III.

IX. Discussion

A. Increasing the Number of Edges in Routing Graph

In BusCast, increasing the number of edges t in $\mathbb{G}'$ implies more flexibility. As t grows, the size of ARS is also linearly increasing, which consumes more communication bandwidth. In a vehicular scenario, where the wireless link quality is very dynamic, long messages may suffer failures. In BusCast, for $\phi = 160$ bits (with security comparable to 1024-b RSA encryption), the length of ARS is $20 \cdot (t + 2) + \omega$ B. Simulation results show that a relatively small t can significantly boost packet forwarding. The typical length of t is several tens. Buses regenerate and verify ARSs without interactions with others. The operations related to ARS are not restricted by the connecting time between buses. Hence, we do not analyze the computation time of ARS in detail.

B. Buses Intrusion Countermeasure

Once TA receives an intrusion warning report from R_i, TA takes the following responses to reset indicators of R_i. First, TA reallocates a new ID for the invaded routes and generates corresponding secret indicators using master key s. Second, it resets all indicators of the route R_i using a secure channel. Third, it notifies other buses and users for ID updating.

TABLE III
COMPUTATIONAL AND STORAGE COMPLEXITY OF ARS

Cost of Construction	Cost of Re-construction	Cost of Verification	Cost of Secret Indicator Generation	Size of ARS	Number of System Parameters
$(t^2+t+2)GM$ $+(t^2-t)GA$ $+2MUL$ $+1EX+1GE$ $+1e+1H_3$ $+1H_2$	$(t+2)GM$ $+4MUL$ $+1GE+1e$ $+1H_3+1H_2$	$(t-1)GM$ $+(t-1)GA$ $+2e+1H$ $+1H_3$	$(t+2)\zeta$ $+\omega$	$1GM$ $+1GA$ $+1H_1$	13

X. ROUTING PERFORMANCE EVALUATION

A. Methodology

Here, we examine the performance of BusCast through trace-driven simulations. We compare BusCast with two alternative schemes.

- *Onion routing*: In this scheme, buses relay a packet according to the predecided shortest path, which is computed on the contact graph of the bus network.
- *Epidemic*: In this scheme, buses exchange every packet whenever they experience a contact. Using this scheme can achieve the shortest end-to-end delay; however, it also generates a tremendously large volume of network traffic.

We consider three metrics to evaluate the performance of our algorithm and the given schemes, including *end-to-end delay*, *delivery ratio*, and *network traffic per packet*.

In the following simulations, we use the same trace data described in Section III. At the beginning of each experiment, we inject 100 packets using a Poisson packet generator. For each packet, the source and destination are randomly chosen within the downtown area. The transmission range of buses is set to 900 m. We use two time periods of trace, namely, off-peak hours (from 2:00 P.M. to 5:00 P.M.) and peak hours (from 5:00 P.M. to 8:00 P.M.) on February 28, 2007 to conduct the simulations. For each simulation configuration, we run the simulation 20 times and obtain the average.

B. Effect of Routing Graph Size

We first examine the effects of the size of routing graph $\mathbb{G}'$ to the delivery performance. For each packet, $\mathbb{G}'$ is generated by picking the k-shortest paths from the source bus to the destination on contact graph $\mathbb{G}$. Duplicate edges between paths are removed. In this set of simulations, $\mathbb{G}$ is constructed in a static way, as described in Section VI-B. Unicast is used to forward packets. We change k from 1 to 20 at an interval of one. In Fig. 8, the red line indicates that increasing k will result in a higher delivery ratio. The delivery ratio increases very fast when k is smaller than 6; after that, the growth becomes slow. It implies that a high cost–performance ratio can be achieved when k is 6. We also verify the *compression ratio of edges* in $\mathbb{G}'$. Denote the number of the edges on path i p_i and the number of edges in $\mathbb{G}$ q. The *compression ratio of edges* is defined as $1-(q/\Sigma_1^k p_i)$. The blue line shows that the compression ratio is larger than 40%, even for very small k, and the compression ratio increases up to about 60% when k is larger than 8. Fig. 8 also plots the end-to-end delay as a function of the number of

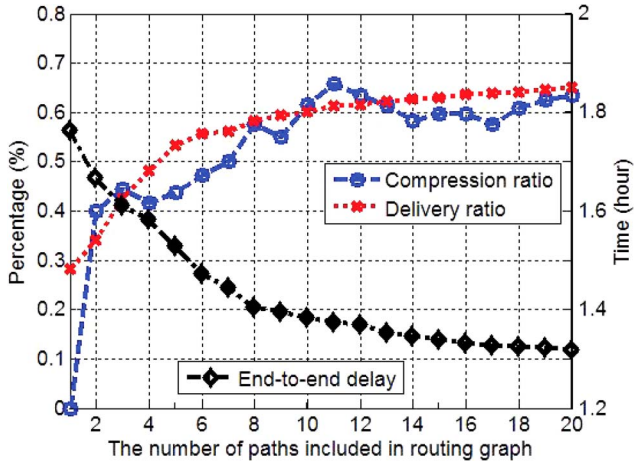


Fig. 8. Delivery ratio of packets and compression ratio of edges under a different number of paths in $\mathbb{G}'$.

k in the routing graph. It is clear to see that, as the number of paths increases, the delivery delay dramatically drops. The average traffic per packet is six hops.

C. Performance Comparisons Under Dynamic Traffic

In this experiment, we compare BusCast with other alternative schemes under dynamic traffic conditions, using both static and dynamic contact graph where the weight $w_{i,j}$ of edge $\hat{e}(v_i, v_j)$ are calculated in different ways.

- In static graph, according to the geographical feature of v_i and v_j, $w_{i,j}$ is proportional to the contact opportunities between v_i and v_j (see the details in Section VI-B).
- In a dynamic contact graph, we use the *average intercontact time of routes* to determine $w_{i,j}$. We refer to the intercontact time as the time elapsed between two successive contacts of two bus routes. To obtain the intercontact time of route v_i and v_j, we first extract all contacts between buses from v_i and v_j, respectively, and sort the contacts in terms of time; then, the intercontact time is computed at the end of each contact as the time period between the end of this contact and the start of the next contact between the same two routes.

We change k from 2 to 20 at an interval of two and conduct the experiments. Figs. 9 and 10 plot the end-to-end delay during off-peak and peak hours, respectively. It can be seen that BusCast can achieve very short delay comparing with onion routing. Epidemic routing has the shortest delay due to flooding guarantees that the optimal path can always be found;

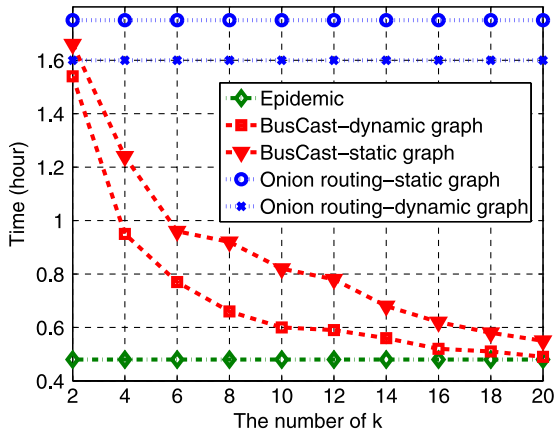


Fig. 9. End-to-end delay during off-peak hours.

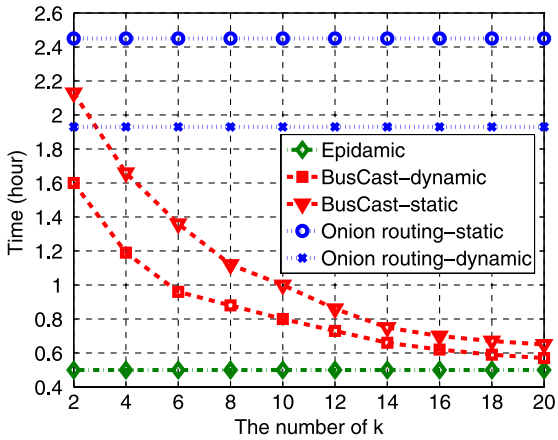


Fig. 10. End-to-end delay during peak hours.

however, it also generates prohibitive network traffic. As k increases, the delay of BusCast is reduced. When k reaches to 20, BusCast can achieve a very close delay comparing with epidemic routing. Meanwhile, BusCast only generate moderate traffic. For example, when $k = 6$, BusCast generates 9.3 hops traffic per packet when using dynamic contact graph during peak hours. It can also be seen that using dynamic contact graph is always better than using static graph, particularly when routing during peak hours.

XI. Conclusion and Future Work

In this paper, we have developed the message delivery scheme BusCast, which ensures both efficiency and users privacy in time-insensitive scenarios. A flexible routing strategy is proposed to adapt highly dynamic changes of bus network topologies. A three-part privacy-preserving mechanism is introduced to ensure anonymous communications. We have demonstrated the efficacy of BusCast through rigorous analysis and extensive trace-driven simulations. For our future work, we intend to investigate different schemes of routing graph generation and to develop shortened ARS to reduce the cost of bandwidth. Furthermore, we will validate our design and study its performance under real complex environments. Improvements will be made based on the realistic studies before it gets deployed in SG.

References

[1] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: traffic data dissemination using car-to-car communication," *Mobile Comput. Commun. Rev.*, vol. 8, no. 3, pp. 6–19, Jul. 2004.

[2] M. D. A. Florides, T. Nadeem, and L. Iftode, "Location-aware services over vehicular ad-hoc networks using car-to-car communication," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1590–1602, Oct. 2007.

[3] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: a popularity aware content sharing scheme in VANETs," in *Proc. ICDCS*, Montreal, QC, Canada, Jun. 2009, pp. 223–230.

[4] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[5] Y. Liu, R. Zhang, J. Shi, and Y. Zhang, "Traffic inference in anonymous MANETs," in *Proc. SECON*, Boston, MA, USA, Jun. 2010, pp. 1–9.

[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second generation onion router," in *Proc. USENIX Security Symp.*, San Diego, CA, USA Aug. 2004, p. 21.

[7] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal reencryption for mixnets," in *Proc. CT-RSA*, San Francisco, CA, USA, Feb. 2004, pp. 163–178.

[8] R. Lu, X. Lin, and X. Shen, "SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *Proc. INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[9] O. Berthold, H. Federrath, and S. KÄopsell, "Web MIXes: A system for anonymous and unobservable Internet access," in *Designing Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, LNCS 2009, H. Federath Ed. New York, NY, USA: Springer-Verlag, 2001, pp. 115–129.

[10] M. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proc. CCS*, Washington, DC, USA, Nov. 2002, pp. 193–206.

[11] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *Proc. S&P*, Berkeley, CA, USA, May 2003, pp. 2–15

[12] R. Zhang, Y. Zhang, and Y. Fang, "AOS: An anonymous overlay system for mobile ad hoc networks," *J. Wireless Netw.*, vol. 17, no. 4, pp. 843–859, May 2011.

[13] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proc. MOBIHOC*, New York, NY, USA, Jun. 2003, pp. 1–12.

[14] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks," *Comput. Commun.*, vol. 28, no. 10, pp. 1193–1203, Jun. 2005.

[15] R. Song, L. Korba, and G. Yee, "AnonDSR: Efficient anonymous dynamic source routing for mobile ad-hoc networks," in *Proc. SASN*, Alexandria, VA, USA. Nov. 2005, pp. 32–42.

[16] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. Shen, "Network coding based privacy preservation against traffic analysis in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 834–843, Mar. 2011.

[17] R. Jansen and R. Beverly, "Toward anonymity in delay tolerant networks: Threshold pivot scheme," in *Proc. MILCOM*, San Jose, CA, USA, Nov. 2010, pp. 587–592.

[18] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Proc. PKC*, Les Diablerets, Switzerland, Jan. 2005, pp. 362–379.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003. Extended abstract in proceedings of Crypto'01, Santa Barbara, CA, USA, Aug. 2001.

Shan Chang (M'08) received the B.S. degree in computer science and technology and the Ph.D. degree in computer software and theory from Xi'an Jiaotong University, Xi'an, China, in 2004 and 2013, respectively.

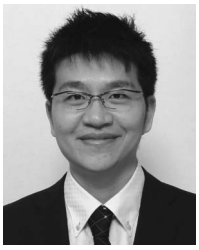
She is currently an Assistant Professor with the School of Computer Science and Technology, Donghua University, Shanghai, China. Her research interests include security and privacy in mobile networks and sensor networks.

Dr. Chang is a member of the IEEE Computer and Communication Societies.

Hongzi Zhu (M'06) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2009.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include vehicular networks, mobile computing, and smart computing.

Dr. Zhu is a member of the IEEE Computer and Communication Societies.



Mianxiong Dong (M'13) received the B.S., M.S., and Ph.D. degrees in computer science and engineering from The University of Aizu, Aizuwakamatsu, Japan, in 2006, 2008, and 2013, respectively.

He is currently an Assistant Professor with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran, Japan. He is also currently a Research Scientist with the A3 Foresight Program (2011–2016), funded by the Japan Society for the Promotion of Sciences, by the National Natural Science Foundation of China, and by the National Research Foundation of Korea. His research interests include sensor networks, vehicular ad-hoc networks, and wireless security.

Kaoru Ota (M'12) received the M.S. degree in computer science from Oklahoma State University, Stillwater, OK, USA, in 2008 and the Ph.D. degree in computer science and engineering from The University of Aizu, Aizuwakamatsu, Japan, in 2012.

She is currently an Assistant Professor with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran, Japan. Her research interests include wireless sensor networks, vehicular networks, and ubiquitous computing.

Xiaoqiang Liu received the B.S. and M.S. degrees in computer science and technology from Harbin Institute of Technology, Harbin, China, in 1990 and 1995 and the Ph.D. degree in control theory and control engineering from Donghua University, Shanghai, China, in 2003.

Since 2009, she has been a Professor with the School of Computer Science and Technology, Donghua University. Her research interests include adaptive information systems and cloud computing.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982 and the M.Sc. and Ph.D. degrees from Rutgers University, New Brunswick, NJ, USA, in 1987 and 1990, respectively, all in electrical engineering.

He is currently a Professor and the University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include resource management in interconnected wireless/wired networks, wireless network security, wireless body area networks, smart grid, and vehicular ad hoc and sensor networks.

Dr. Shen is a Distinguished Lecturer of the IEEE Vehicular Technology and Communications Societies. He is a Fellow of the Canadian Academy of Engineering and the Engineering Institute of Canada. He is a registered Professional Engineer in Ontario, Canada.