Peer-to-Peer Networking and Applications (2021) 14:1475–1487
https://doi.org/10.1007/s12083-020-01045-7

Exploiting peer-to-peer communications for query privacy preservation in voice assistant systems

Bang Tran[1] · **Xiaohui Liang**[1]

Received: 20 August 2020 / Accepted: 30 November 2020 / Published online: 7 January 2021

Abstract

Voice assistant system (VAS) is a popular technology for users to interact with the Internet and the Internet-of-Things devices. In the VAS, voice queries are linked to users' accounts, resulting in long-term and continuous profiling at the service provider. In this paper, we propose a VAS anonymizer aiming to mix the queries of the VAS users to increase the source anonymity. The VAS anonymizer is equipped with a pattern-matching scheme, which allows VAS devices to find effective peer relays without disclosing their query patterns. Furthermore, the VAS anonymizer is equipped with anonymity evaluation modules for evaluating real-time single query, thus reducing the risk of pattern violation at the relays. Both the requester and the relay will evaluate the real-time query based on the resulting anonymity. Only if the anonymity evaluations at both requester and relay are positive, the query will be sent to the service provider via the relay. The VAS anonymizers at VAS devices coordinate the query uploading such that the sources of the queries are anonymized, and the service provider is unable to link the voice queries to individual users. In the experiments using our customized VAS devices and the Amazon Cloud servers, the computation and communication overhead of the matching scheme is shown to be efficient, and the anonymity evaluation modules are shown to be effective in protecting the privacy of the requesters and the relays.

Keywords Voice assistant systems · Peer-to-peer communications · Query privacy · Source anonymity

1 Introduction

Voice assistant system (VAS) becomes increasingly popular for enabling voice interaction with the Internet and the Internet-of-things (IoT) devices. Juniper Research estimates about 3.25 billion voice assistants are in use in 2019, and about 8 billion voice assistants will be in use in 2023 [1, 2]. Common VAS devices like Apple Siri, Amazon Alexa, and Google Assistant have been integrated into many consumer mobile devices and vehicles, and deployed in homes, university dorms, and hotels [3, 4]. VAS is typically a centralized system where voice analytics and machine learning are run at centralized VAS service providers (VSPs) [5]. While the AI at VSPs is kept evolving with users' voice data, the voice data exposure to VSPs raises serious privacy concerns [6]. Recent privacy efforts aim to *minimize the characteristic inference* from voice analysis. For example, offline speech recognition processes voice data locally for better privacy and less communication cost [7–9]. Federated learning enables participating clients to train shared machine learning models while keeping their data locally [6, 10, 11]. A local differential privacy scheme randomizes the data before uploading, so the server never receives raw data [12, 13]. While the data is locally processed to be a minimum upload for needed services, *anonymizing data source* is a complementary approach to enhance user privacy. Conventional network-level anonymizers, such as TOR [14] and VPN [15], used for preventing traffic analysis attacks, do not effectively anonymize the source of user's voice data from the VSP; the user's login behavior enables the VSP to link the voice queries to the same user regardless of the device-level and network-level anonymization techniques. An example

This article is part of the Topical Collection: *Special Issue on Privacy-Preserving Computing*
Guest Editors: Kaiping Xue, Zhe Liu, Haojin Zhu, Miao Pan and David S.L. Wei

✉ Xiaohui Liang
xiaohui.liang@umb.edu

Bang Tran
bang.tran001@umb.edu

[1] Department of Computer Science, University of Massachusetts Boston, Boston, MA, USA

is Google MyActivity, linking voice queries to Google accounts [16].

We aim to minimize the data linkability while maintaining the effectiveness and efficiency of the VAS. Specifically, we plan to exploit the **peer-to-peer communications** among the VAS devices, to anonymize the source of the queries without changing user behavior and service model. The anonymizer provides anonymity protection as follows. *With the anonymizer*, the query of a requester (a VAS device) is sent via peer-to-peer communications to a relay (another VAS device), which then uploads the query to the VSP using the relay's account. The response from the VSP relevant to the query is sent back to the requester. In this case, the source of the query is hidden from the VSP, and the relay's real queries are mixed with the requester's query. Note that, such a VAS anonymizer applies to the VAS services that can be performed purely based on the query content regardless of the user account (named anonymizable services). Our preliminary results show that 58% of the VAS services are anonymizable (as shown in Section II). For these services, the proposed anonymization technique minimizes the data linkability risk without changing the current user behavior and service model. The design of the proposed anonymizer faces a set of new research challenges.

Anonymity Our anonymity objectives are (i) the VSP is uncertain about the source of the query, (ii) the VSP cannot identify the relay's real queries from the mixed queries, and (iii) the relay's pattern does not significantly change after the relaying behavior. These objectives are shown in Fig. 1. To achieve these objectives, the similarity evaluation between two query patterns and the similarity evaluation between a query and a query pattern are necessary components.

We envision that in our system the requester and the relay will be able to match their query patterns without disclosing the details of their queries. After a requester and a relay have their query patterns matched, for a real-time query generated at the requester, it may still significantly deviate from the query patterns used for the matching. The deviated query should not be sent to the VSP. Specifically, if the requester negatively evaluates the query, the query should not be sent to the relay; if the query is sent to the relay, but the relay negatively evaluates the query, the relay should not send the query to the VSP. The evaluations by the requester and the relay on the real-time query should be effective so as to protect the anonymity of both the requester and the relay.

Data exposure and service delay The anonymizer makes use of peer relays run by volunteers, who gain temporary access to the content of the query. For a period of time, the relay may collect the query patterns of the requester, incurring data exposure risk. The requester needs to evaluate such risk and change relay if necessary. Other than the data exposure risk, the peer-to-peer communications introduce additional delay to the VAS services. Therefore, the workload of the relays and the peer-to-peer communication delay affect the VAS service quality and should be considered in the matching process.

We propose a VAS anonymizer, consisting of a pattern matching scheme and anonymity evaluation modules. The matching scheme enables the requester to find an effective relay that has the most similar query pattern to itself. The evaluation modules are further used to evaluate the similarity of a real-time query and the query pattern such that the query to be sent by the relay does not significantly deviate from the relay's query pattern. Specifically, we first define a new data structure to represent the query pattern, which contains the application type, the usage frequency, and the occurrence time. We choose these factors because the query and response in the VAS need to be delivered in real-time; if the requester and the relay send the same query at significantly different time or use it with significantly different frequency, the query pattern at the relay after including the query from the requester would change significantly, resulting in less anonymity. We aim to efficiently find the most effective relay for a requester while the query patterns of both the requester and the relay will

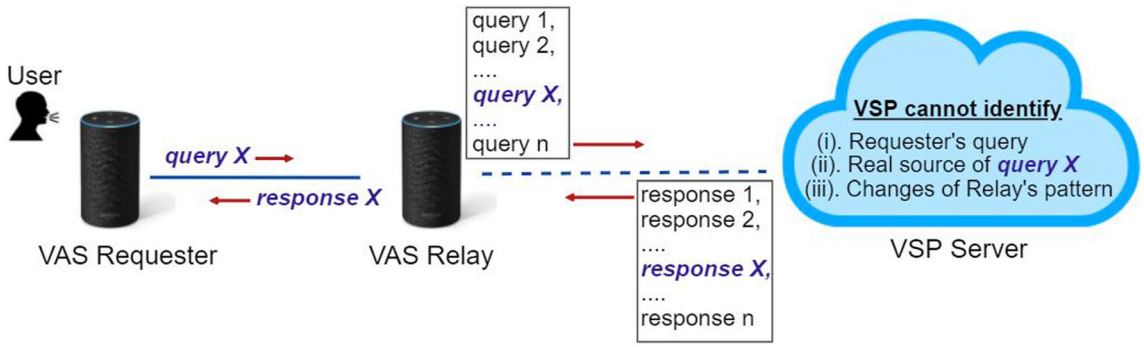


Fig. 1 The anonymity objectives of VAS service

not be disclosed in the first place for privacy preservation. The contributions of this paper are four-folds.

First, we propose a novel VAS anonymizer that anonymizes the source of the query in the VAS. The VAS anonymizers coordinate and mix the queries from multiple VAS users' devices such that the VSP is unable to link the queries to individual users for long-term profiling. The proposed anonymizer applies to 58% of the existing VAS applications.

Second, we propose a privacy-preserving pattern matching scheme, which enables a semi-trusted server to help find the most effective relay for a requester. The matching is conducted on unique data structures preserving the application, frequency, and occurrence time information of the queries, and it does not disclose the details of the queries to the server and other peers.

Third, we propose anonymity evaluation modules on both the requester and the relay to evaluate the real-time query. If a query significantly deviates from the query pattern of the requester, the requester will not send the query to the relay; if the query is sent to the relay, but significantly deviates from the relay's query pattern, the relay will not send the query to the VSP.

Fourth, we implemented and evaluated the VAS anonymizer using the query patterns of real users, obtained from the Google MyActivity. We showed the matching scheme could be efficiently performed at a semi-trusted server in realistic settings while preserving the privacy of the query patterns. We also showed the evaluation modules ensure anonymity effectiveness while lowering communication cost.

Since the AI techniques, the machine's computation ability, and the algorithms for natural language processing have become more advanced in recent years, the users' voice data exposed to the VSPs would raise serious privacy concerns. We envision that our study on exploring the anonymization of queries among users via peer-to-peer communication will produce a significant impact.

2 VAS services

Based on the smart speaker consumer adoption report in March 2018 [17], the top use cases of voice assistant at smart speakers are listed in Table 1. Columns "Tried", "Daily", and "Monthly" represent the percentages of smart speaker users using the service at least once ever, once daily, and once monthly, respectively. "Local" means the services can be implemented on local devices with local networks. "Anony" means the services can be anonymized using the proposed anonymizer. We find that the proposed anonymizer applies to 10 of 17 services (58%), potentially enhancing the anonymity of the voice query. Four services 4, 6, 14, and 15 can be implemented on local devices with local networks. One common characteristic of the local services is that the devices and data belong to the same user who creates and uses them. The local services can be completely implemented locally and have no privacy

Table 1 In-market most popular VAS services [17]

	Services	Tried	Monthly	Daily	Local	Anony
1	Ask a question	91%	72.9%	33.3%	×	✓
2	Listen to streaming music service	89.5%	76.2%	41.9%	×	✓
3	Check the weather	85.2%	69.1%	41.4%	×	✓
4	Set a timer	71.4%	51.8%	24.1%	✓	
5	Listen to radio	68.8%	47.6%	25.5%	×	✓
6	Set an alarm	65.7%	48%	25.3%	✓	
7	Listen to news / sports	58.1%	39.4%	14.8%	×	✓
8	Play game or answer trivia	52.3%	31.2%	11.1%	×	✓
9	Find a recipe or cooking instructions	49.5%	26.5%	5.1%	×	✓
10	Use a favorite skill or assistant app	46.5%	29.8%	14.7%	×	✓
11	Check traffic	41.2%	25.8%	7.7%	×	✓
12	Call someone	40.7%	22.7%	10.3%	×	×
13	Listen to podcasts and other talk formats	40.7%	24.1%	10.1%	×	✓
14	Control smart home devices	38.1%	29.9%	20.8%	✓	
15	Access my calendar	35.1%	19.6%	6.2%	✓	
16	Message someone	34.2%	17.9%	8.2%	×	×
17	Make a purchase	26%	11.5%	2.1%	×	×

concerns. 10 of 17 services are *anonymizable services*. If the queries are self-contained, the anonymizer applies to them. There are some exceptions. Services 3 and 11 use the location information; the anonymizer needs to ensure the relay and the requester imply the same location. When a third-party app is activated (service 10), such as "play classical music at Pandora", the anonymizer needs to ensure both the requester and the relay have access to the app. If the queries are not self-contained, the users implicitly allow the VSPs to access their profiles for completing the queries.

3 System model

In this section, we introduce four system entities as shown in Fig. 2 and four primary design goals.

3.1 System entities

VAS service provider (VSP) Typical VAS service providers are Amazon Voice Services or Google Assistant Services. The VSPs process users' voice queries using speech recognition and natural language processing techniques, and then accurately respond to the queries [18].

A **semi-trusted server (STS)** in a "honest-but-curious" mode is trusted for accurately running the proposed matching scheme, but not trusted for accessing any privacy-sensitive query patterns of the requester and the relay. The VAS devices are registered to the semi-trusted server and periodically involved in the matching process to update their relays based on the updated query patterns and data exposure risk. The STS then securely communicates to the VAS devices and publishes authentic information. The tasks of the STS include i) sharing information in a public and authentic channel; ii) running the matching scheme; iii) acting as a router to enable the peer-to-peer communication among VAS devices in different local networks.

VAS devices as requester have three components: (i) An offline speech recognition module. Recent study showed the module can be effectively developed using Amazon Mechanical Turk for collecting voice samples and data augmentation [19] for simulating noisy and reverberant conditions in virtual rooms of different sizes and random microphone and speaker locations [8]. (ii) A matching module. It communicates with the STS periodically aiming to find an effective relay based on the recent query pattern. (iii) An anonymity evaluation module evaluates the similarity of the real-time single query with the query pattern used for the previous matching. If the query significantly deviates from the pattern, the query will not be sent to the relay.

VAS devices as relay mix the queries from multiple requesters while maximizing the anonymity and minimizing the communication overhead. It has three components: (i) A registered VSP interface to upload text-based queries to the VSP. (ii) A matching module to communicate with the STS periodically aiming to provide relaying service to the requester with a similar query pattern. (iii) An anonymity evaluation module to evaluate the real-time single query from the requester for its consistency with the current query pattern. If the query significantly deviates from the pattern, the query will not be sent to the VSP.

As described above, the proposed VAS anonymizer has two main components, a matching module and an anonymity evaluation module. With the help from the STS, the matching module enables a requester and a relay to be matched based on their query patterns. The anonymity evaluation modules further ensure the anonymity protection by evaluating the real-time query, as any new query may deviate from the query patterns previously used in matching. The anonymity evaluation module at the requester rejects the deviated query for saving communication cost, while

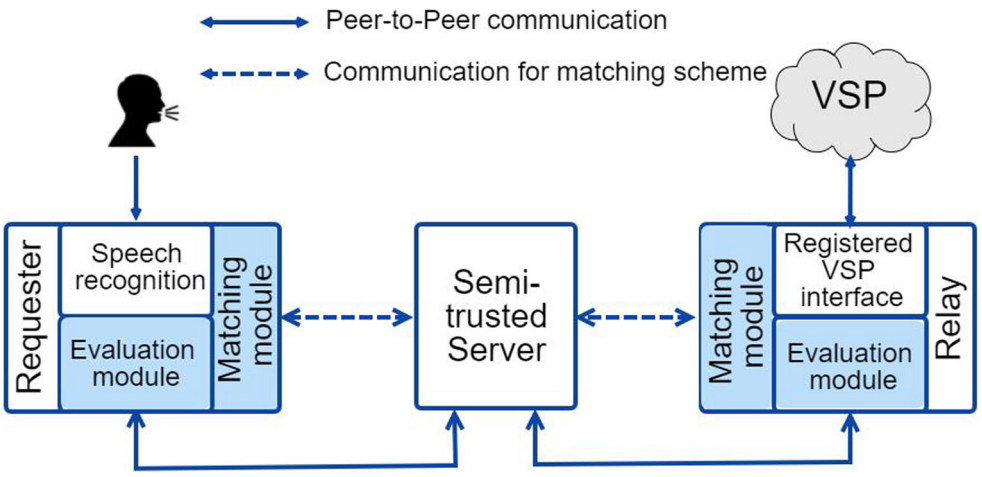


Fig. 2 System model

the module at the relay rejects the deviated query for maximizing anonymity effectiveness.

3.2 Design goals

Our design goal is to implement an effective VAS anonymizer at VAS devices such that the queries from different users' accounts are mixed and anonymized from the VSP's perspective. Specifically, we aim to achieve the following goals.

- **Effective matching.** Our matching scheme aims to enable a requester and a relay to be matched based on the similarity of their query patterns. The similarity is calculated based on two unique data structures representing two query patterns, with the consideration of the application, frequency and occurrence time information. The time factors are important because the queries and responses are delivered in real-time.
- **Pattern privacy.** The query pattern is highly privacy-sensitive. In the matching scheme, both the requester and the relay's query patterns are neither disclosed to the STS nor shared with other peers. The matching process at the STS outputs the relay that most effectively matches with the requester without accessing their patterns.
- **Anonymity effectiveness.** We consider that a real-time query from a requester may deviate from the requester's past query pattern. In this case, if a requester sends a deviated query to the previously-matched relay, it may incur a risky situation of pattern violation at the relay. Our VAS anonymizer incorporates the anonymity evaluation module to keep the anonymity protection effective and eliminate the unnecessary communication cost on possibly rejected queries.
- **Efficiency.** Our scheme should be computation- and communication-efficient. Our system runs the intensive matching task on a semi-trusted server rather than the VAS devices. The computation at the VAS devices, and the communication between the VAS devices and the server, should be all efficient considering that the computational and communication capabilities of the VAS devices are limited.

4 Proposed scheme

In this section, we first propose a pattern matching scheme where a semi-trusted server helps a requester to find an effective relay with similar query patterns. We then present the anonymity evaluation modules at both the requester and the relay to ensure effective anonymity protection.

4.1 Pattern matching

4.1.1 Data structure of query pattern

We consider the VAS system has n applications $\mathcal{A}_i$ for $1 \leq i \leq n$. We count the number of times a user uses application $\mathcal{A}_i$ in a timeslot. To compare the patterns of two users, we consider the pattern in a consecutive k timeslots. To be comparable, the length of the query pattern, i.e., k timeslots, can be a day, a week, or a month. The pattern is defined as an $(n \times k)$-dimensional vector $h = \left(x_{i,j}\right)$ where $1 \leq i \leq n, 1 \leq j \leq k$, and $x_{i,j}$ is the number of times $\mathcal{A}_i$ is used during j-th timeslot. $h = (\underbrace{x_{1,1}, x_{1,2}, \ldots, x_{1,k}}_{\mathcal{A}_1}, \ldots, \underbrace{x_{n,1}, x_{n,2}, \ldots, x_{n,k}}_{\mathcal{A}_n})$ The vector can be large, resulting in a significant computation and communication overhead. However, considering the VAS users have a relatively regular VAS usage behavior, they can be grouped using a full vector in the first round and then matched using a smaller-size vector for only common applications. Once a user exploits a new VAS application, the full vector can be used again to move the user into a new group.

4.1.2 Similarity calculation

Our system sets up a semi-trusted server for calculating the similarity. A simple solution is the server receives the vectors from both the requester and the relay. The server calculates the distance of any two vectors for the similarity. Denote two vectors as $h(x) = (x_{1,1}, \ldots, x_{n,k})$ and $h(y) = (y_{1,1}, \ldots, y_{n,k})$ The similarity of the two vectors h_x and h_y is:

$$\|h(x), h(y)\| = \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{(x_{i,j} - y_{i,j})^2}{y_{i,j}} \quad (1)$$

The similarity calculation implies that only if the same application used in the same timeslot with a similar usage frequency, the score is kept small. In addition, we consider relative difference instead of absolute difference, i.e., for the same difference, when $y_{i,j}$ increases, the contribution to the similarity decreases, e.g., (9, 10) is more similar than (0, 1).

4.1.3 Privacy-preserving pattern matching

We further propose a privacy-preserving pattern matching scheme, which does not disclose the original vectors to the semi-trusted server but enables the server to calculate the similarity score as defined in the previous section. The implementation includes four steps, initialization, data upload by requester, data upload by relay candidate, and similarity calculation.

Initialization The semi-trusted server helps share information between requesters and relay candidates. Consider a relay candidate has Diffie-Hellman (DH) [20] parameters (a, g^a), and a requester has DH parameters (b, g^b) where g is shared by the server. If a relay candidate is available, its g^a is included in the relay candidate list. If a requester makes a request, its g^b is included in the requester list. Both lists are tagged with a time stamp and published by the server in real-time. Without the interaction between the requester and the relay, they successfully share a key g^{ab}. Note that, as the server acts as a router between the requester and the relay, it knows the communication delay of server-to-requester and server-to-relay. Thus, it can estimate the communication delay of requester-to-relay by simply adding the two delays. In addition, the server knows how many requesters the relay is matched with and the amount of workload on the relay. The server may prefer to choose the relay candidate with a shorter delay (specific to a given requester) and with less workload.

Data upload by requester The requester extends $h(y) = (y_1, \cdots, y_N)$ to a $(N+3)$-dimensional vector

$$\bar{h}(y) = \left(1, \frac{1}{y_1}, \cdots, \frac{1}{y_N}, R', 1\right) \tag{2}$$

where R' is a random number. The requester then chooses an invertible $(N+3)\times(N+3)$ matrix Q, and securely shares it with the relay candidates using the DH key g^{ab} via the server. The requester calculates $g(h(y)) = \bar{h}(y) \times Q^{-1}$ and uploads it to the server as well. Note that Q can be repeated used until the requester decides to discard it. The use times of Q will be discussed in Section V.

Data upload by relay candidate The relay candidate extends $h(x) = (x_1, \cdots, x_N)$ to a $(N+3)$-dimensional vector

$$\bar{h}(x) = \left(-\sum_{i=1}^{N} 2x_i - R, x_1^2, \cdots, x_N^2, 1, R\right) \tag{3}$$

where R is a random number. The relay candidate then decrypts the matrix Q from the requester using the shared secret g^{ab}. The relay candidate calculates and uploads $g(h(x)) = Q \times \bar{h}(x)^\top$ to the server.

Similarity calculation The server receives $g(h(x)) = Q \times \bar{h}(x)^\top$ and $g(h(y)) = \bar{h}(y) \times Q^{-1}$, and calculates sim' for each relay candidate and the requester as follows:

$$\begin{aligned} sim' &= \bar{h}(y) \times Q^{-1} \times Q \times \bar{h}(x)^\top = \bar{h}(y) \times \bar{h}(x)^\top \\ &= \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{y_i} + R' - \sum_{i=1}^{n} y_i \\ &= sim + R' - \sum_{i=1}^{n} y_i \end{aligned} \tag{4}$$

The server calculates sim' for all relay candidates and returns the one with the lowest score sim' to the requester. Since the requester knows R' and $\sum_{i=1}^{N} y_i$, the requester then calculates the original similarity sim. If $sim < th$, the requester considers the best-matched relay is in good matching condition. If the best-matched relay is not in a good condition, i.e., $sim \geq th$, the requester may wait for a while and rerun the matching scheme with the server.

After the matching scheme, the requester acknowledges the server its decision on the best-matched relay. The requester will send any anonymized query to the server with the relay's identify g^a. The server acts as a router to forward the query and the response between the requester and the relay. The communication is end-to-end authenticated and encrypted using DH key g^{ab}. As the matching is conducted based on their VAS application query patterns, before sending the real-time query, the requester needs to ensure the query fits their query patterns. In the following, we propose the anonymity evaluation modules to ensure the anonymity effectiveness for the real-time query.

Since the semi-trusted server knows the history of matching between pairs of VAS devices, a module supporting machine-learning solutions can be added to boost the performance of matching process at the STS from the rejecting rate between a pair of devices. e.g., a supervised regression module can help extract a subset of relay candidates from the whole list of available candidates. We plan to expand this idea in our future work. In this paper, we still keep calculating as shown on equations from (1) to (4) on every relay candidates.

4.2 Evaluation modules

The goal of the proposed VAS anonymizer system is to ensure that the relay only uploads real-time queries that are not deviated from the relay's pattern, thus achieving effective anonymity protection. We propose two evaluation modules for a requester and a relay. The requester evaluates the real-time query on its side before sending it to the relay, and the relay evaluates the real-time query before uploading it to the VPS server. In the following, we first introduce how to evaluate the similarity between a query and a query pattern and then present the evaluation modules in detail.

A single query can be represented as a k-dimension vector $\mathcal{Y} = (y_1, \ldots, y_k)$ where $y_t = 1, y_i = 0$ for $i \neq t$. t is the timeslot index in the pattern period when the query occurs. We consider the query pattern of a certain application as $\mathcal{X} = (x_1, x_2 \ldots, x_k)$ where x_i is the number of the invocations in timeslot ith, $i \in [1, k]$. We define $\mathcal{F}_{eval}(\mathcal{X}, \mathcal{Y})$ to measure the similarity of a query and a query pattern as:

$$\mathcal{F}_{eval}(\mathcal{X}, \mathcal{Y}) = 1 - \frac{1}{\beta^z}; z = \sum_{i=1}^{k} \frac{x_i}{\alpha^{|i-t|}}. \quad (5)$$

where t is the timeslot index when the query $\mathcal{Y}$ occurs, x_i is the number of invocations at i-th timeslot in the query pattern $\mathcal{X}$, and (α, β) are two adjustable positive real numbers.

The value range of the function $\mathcal{F}_{eval}(\mathcal{X}, \mathcal{Y})$ is $[0, 1)$. When there is no invocation of the application in the query pattern, the function returns 0, indicating the most dissimilarity. When the number of invocations increases, the function's output value increases. We use two adjustable parameters α, β to weigh the contributions of the invocation at different timeslots. In the query pattern, the timeslot closer to t has more contribution than one faraway from timeslot t. If an invocation happens at the timeslot t, the weight at its is maximum 1. The weight will significantly decrease as the timeslot distance increases, as shown on the right of Fig. 3. In addition, the invocations' contributions decrease as the total number of the invocation increases, as shown on the left of Fig. 3. The maximum output value of the function $\mathcal{F}_{eval}(\mathcal{X}, \mathcal{Y})$ is 1.

Requester's anonymity evaluation When a requester generates a query at a timeslot t, it applies the anonymity evaluation module to measure if this query deviates from its query pattern used in the previous matching. If the evaluation result is negative, the requester considers the query possibly deviates from the relay's pattern, because the relay's pattern is similar to its pattern. Suppose that the requester has a vector $\mathcal{Y}$ representing the query and a vector $\mathcal{X}$ representing the pattern, which is used to find the most effective relay in the matching. The requester calculates $\mathcal{F}_{eval}(\mathcal{X}, \mathcal{Y})$. If the output value is lower than a chosen threshold, the requester does not send the query to the relay. The requester may directly upload the query to the server or rerun the matching scheme for finding another relay. If the output value is higher than the threshold, the requester sends the query to the relay. Note that, the early rejection at the requester helps lower the communication costs.

Relay's anonymity evaluation When the relay receives a query from the requester, the relay applies the anonymity evaluation module to measure if the query deviates from its most recent pattern. Suppose the relay has a vector $\mathcal{Y}$ representing the received query and a vector $\mathcal{X}$ representing its most recent query pattern. The relay calculates $\mathcal{F}_{eval}(\mathcal{X}, \mathcal{Y})$. If the output value is lower than a chosen threshold, the relay rejects uploading the query to the VSP. If the output value is higher than the threshold, the relay uploads the query to the server. Note that, both the requester and the relay record the reject rate. If a query is rejected, the requester may decrease the evaluation threshold in a hope to lower the likelihood of being rejected again. When the reject rate is considered non-tolerable, the requester and the relay inform the semi-trusted server to rerun the matching scheme.

5 Privacy discussion

Our privacy goal is to prevent the disclosure of the privacy-sensitive query patterns of individual users to the

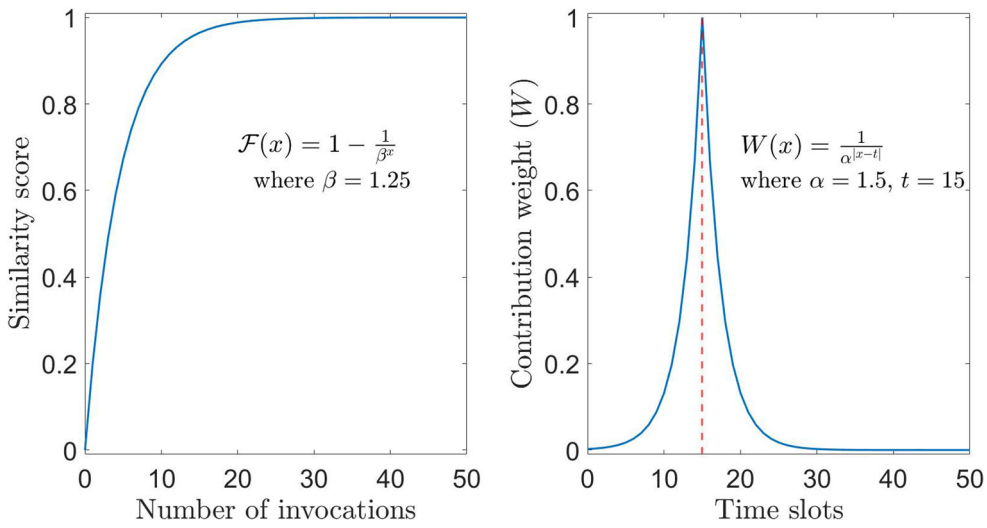


Fig. 3 Evaluation function and Effective weight changes

semi-trusted server while enabling the similarity calculation by the server. From the security perspective, we cannot prevent the server from accessing the query patterns if the server launch a man-in-the-middle attack by putting its own public key $g^{a'}$ into the relay candidate list for obtaining Q. We cannot prevent the collusion attacks between the server and the relay either. A fundamental problem is that anyone can be a relay candidate without restriction, and a solution is to introduce certificate authorities, which is beyond the paper scope.

If the semi-trusted server is "honest-but-curious", we can prove that the server cannot derive any information about the users' query patterns from the uploaded data as follows. As shown in our scheme, a requester and multiple relay candidates may use the same matrix Q to randomize their vectors. Let us assume the total number of Q used in multiplications as m_Q. At the server end, the total number of unknown parameters in these multiplications is $m_p = (N + 1)*m_Q+(N+3)^2$, and the total number of equations is $m_e = (N + 3) * m_Q$. As such, to prevent the $(N + 3)$-vector from being derived by the server, we need to ensure $m_p - m_e \geq N + 3$, i.e., $m_Q \leq \frac{1}{2} \cdot (N^2 + 5N + 6)$. If $N = 216$ as in the previous example, the maximum number of multiplications by the requester and the relay is 23,000. As long as the number of multiplication is less than this upperbound, the server cannot derive any information about the vector. As the requester knows the number of multiplications that have been done, the requester can always choose a new matrix for security before the maximum is reached.

The anonymizer we proposed in this paper employs heuristic functions to estimate the similarity scores between two query patterns and between a real-time query and a query pattern. The calculation of the similarity scores does not require intensive computation efforts at local VAS devices. However, VSP server is computationally powerful and can profile users based on not only their query patterns but also the semantic and syntactic content of their queries. In the future, we plan to enhance our anonymizer by integrating semantic and syntactic analysis in the matching and evaluation modules. One idea is to generate fake queries that are semantically and syntactically similar to the real queries so as to enhance the anonymity protection for the requester and the relay [21].

6 Performance evaluation

In this section, we describe our experiment setup, present the evaluation results of the proposed pattern matching scheme, and present the evaluation results of the proposed anonymity evaluation modules.

6.1 Experiment setup

We used onboard microcomputers RaspberryPi 3 as VAS devices, which run Alexa Voice Service (AVS) Device SDK [22] to integrate with the features and functions of VAS services. As shown in Fig. 4, for the semi-trusted matching server, we first used a local computer server and then tested a remote Amazon server [23], both running Ubuntu 18.04 [24]. We adopted two network environments: i) we connected the customized VAS devices and the local server to a Wi-Fi router TP-LINK N450 [25], forming a local network; and ii) we connected the customized VAS devices to the remote Amazon server, forming a realistic network environment. For enabling VAS services at each customized VAS devices, we registered Google accounts and exploited Google assistant system. In the following, we first describe the evaluation results on the performance of the matching scheme, then provide preliminary performance result on the anonymized VAS services, and finally discuss other performance issues.

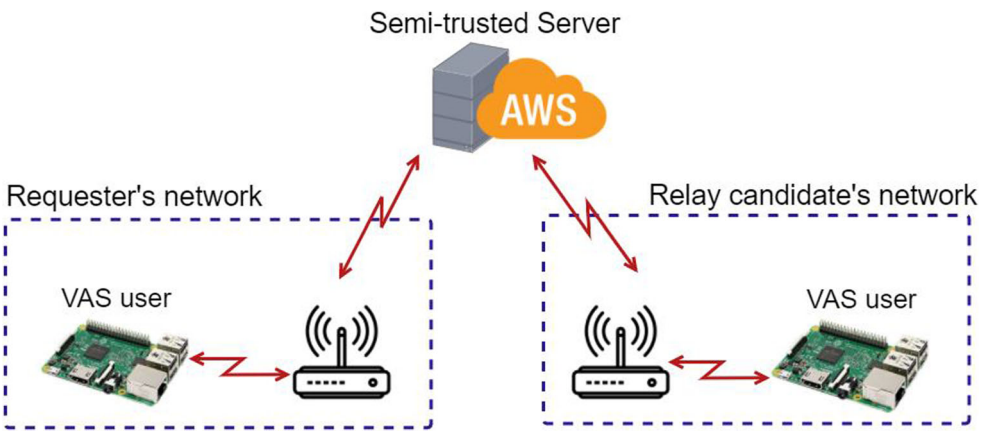


Fig. 4 Experiment model for matching scheme

6.2 Pattern matching evaluation

The proposed pattern matching scheme was evaluated with the settings of the vector sizes (120, 240, 360, 480) and the numbers of relay candidates (10, 20).

Computation overhead Figure 5a, b, c and d report the computation time spent by the requester, the relay, the local server, and the Amazon server [23] on the matching scheme, respectively. We observe that when the vector size increases, the computation costs are significantly more. The maximum vector size in our experiment is 480, i.e., 8 days of 4-hour slot for 10 applications. However, in reality, the size can become bigger for more fine-grained patterns, shorter time slots, and longer duration. As the matching scheme is not needed in real-time, it can be automatically run in the midnight when the VAS devices are much rarely used, and thus the computation is not a problem here. In addition, as we mentioned in Section 4, the VAS users often have a fixed pattern of what and how the VAS applications are used. Thus, the matching scheme can be run in multi-round. The first round uses a full vector to group users based on the used applications; the second round uses a smaller vector only to include the common applications. In this case, the computation overhead can be largely reduced. When we increase the number of relay candidates, we observe the computation on the requester, the local server, and the Amazon server all increase. This is because the requester needs to share the matrix with each relay securely and the server needs to calculate the similarity score for each relay. Comparing the local server and the Amazon server, we found that the computation takes < 0.1s and the Amazon server runs faster.

Communication overhead Figure 6a and b report the whole matching processing time for the local server and the Amazon server, respectively. The time includes the computation time at requester, server and relay, and the communication time between the requester/relay and the server. As we can see, if the matching scheme is run without the privacy requirement, the computation costs at the requester and the relay are none because they directly send their vectors to the server. As such, the matching processing time without privacy is < 6s. If the privacy is required, the matching process time is slightly more than the computation time on the requester. In other words, the delay caused by the communication overhead is negligible compared to the delay caused by the computation overhead. In addition, when comparing the local server and the Amazon server in the same conditions, the Amazon server takes less time on computation, the time on communication between the requester/relay and the Amazon server is relatively more.

6.3 Anonymity module evaluation

The anonymity evaluation modules on both requester and relay are designed to prevent a deviated real-time query to be sent by the relay. To evaluate the effectiveness of anonymity evaluation modules, we used two onboard microcomputers RaspberryPi 3 as VAS devices. The

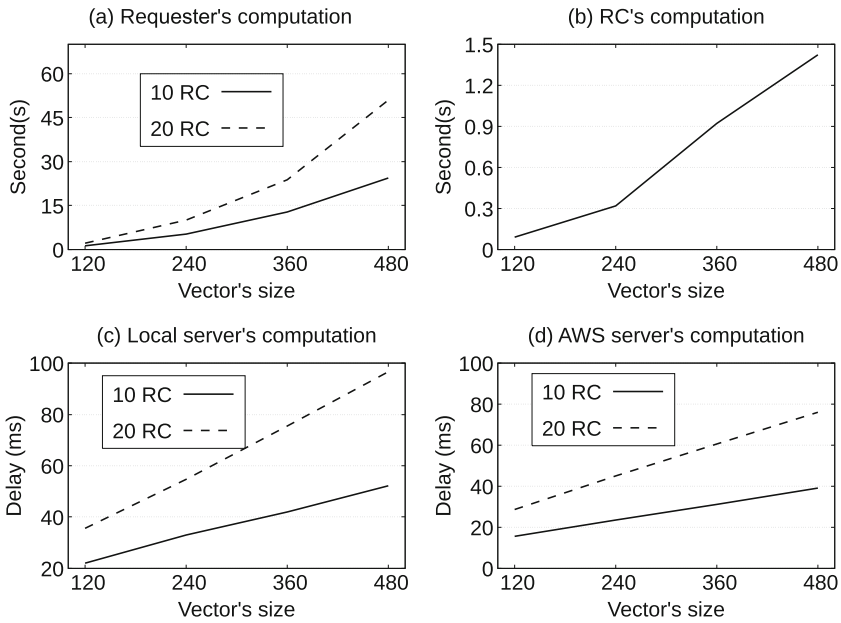


Fig. 5 Matching computation (NP = Non-Privacy, RC = Relay Candidate)

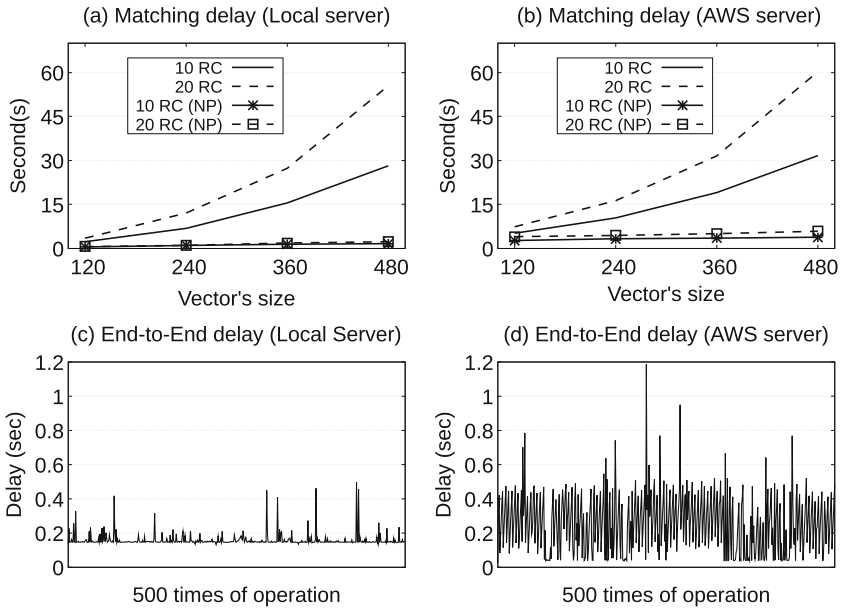


Fig. 6 Communication delay (NP = Non-Privacy, RC = Relay Candidate)

requester and the relay are matched because of their similar query patterns. We generated random queries in different time slots at the requester, which then send them to the relay. The relay will run its evaluation process and informs the requester for every rejected request. The main goal of this evaluation is to measure the reject rate at relay. We will run the experiment in two scenarios (i) the requester sends his queries directly to the relay without running an evaluation process at the requester; and (ii) the requester sends a set of positively-evaluated queries (after running evaluation process) to the relay. The results are shown in Table 2. The results prove that without running the evaluation process at the requester, the reject rate at the relay is two-times higher than the rate when running the evaluation process at the requester. When choosing different vector sizes 120, 240, 360 and 480, the accept rates at relay are 85%, 81%, 78% and 70% respectively in the first scenario. These rates increase to 95%, 92%, 88% and 84% in the second scenario. It means that the evaluation process at requester works effectively to preserve anonymity and lower the communication cost.

6.4 Other performance discussion

Relay trustworthiness Our anonymizer shifts the long-term trust on the service provider to the short-term trust on the multiple distributed relays. We consider relay trustworthiness with two factors. i) The relay's performance can be evaluated at the semi-trusted matching server. For example, the server knows how many requesters are matched with the relay in the past, and it also knows the number of queries and responses the relay transmits for others. These statistics can be seen as the relay trustworthiness. A more trustworthy relay is more likely to fulfill the queries and achieve better performance. ii) The relay's trustworthiness can be evaluated by the requester.

Table 2 Reject rate at relay

—Pattern vector size	Received queries	Rejected queries	Reject rate—
(a) Scenario 1: Without requester's anonymity evaluation			
120	10^5	14873	14.87%
240	10^4	1893	18.93%
360	10^3	216	21.6%
480	10^3	302	30.2%
(b) Scenario 2: With requester's anonymity evaluation			
120	10^5	5127	5.13%
240	10^4	782	7.82%
360	10^3	120	12.0%
480	10^3	164	16.4%

If the number of queries of the requester sent to the same relay exceeds a threshold, the requester may prefer to choose a different relay for a less data exposure risk. The matching design incorporating relay trustworthiness will be considered as future works.

Priority of others' queries The VAS devices are mainly used by their owners. A recent usage report in Table 1 shows 33.3% of the smart speaker users ask a question every day. In other words, the VAS is often in an inactive status. We envision the inactive VAS devices act in a relay role. However, the relay may face a situation that it receives self queries and others' queries to be finished at the same time. To avoid compromising the VAS experience, our anonymizer will block any external queries if the VAS is currently used for self queries. Therefore, the VAS performance is not affected at all by the anonymizer and the potential denial of service attacks.

7 Related works

Many studies focus on privacy preservation of VAS devices and speech recognition. Pathak et al. [26] proposed frameworks which aims to preserver privacy of conventional speeches by computing various operations via secure operations such as secure multiparty computations, additive secret sharing, and secure logsum. These techniques suffer from practical limitations due to their dependence on computationally expensive cryptography. Gao et al. proposed a solution that uses ultrasound jamming to address stealthy recording [27]. Hadian et al. proposed an encryption scheme preserves the privacy of voice data in mhealth system [28]. Qian et al. utilized the keyword substitution technique to sanitize the voice input contents before sending it to a cloud server [29]. They further extended the idea by designing a heuristic algorithm that personalizes the sanitization for speakers to restrict their privacy leak [30]. Glackin et al. use a neural network to encode the symbolic audio data before uploading to the server, which then employs searchable encryption over the speech content [31]. Besides, Li et al. developed a multi-keyword search scheme over encrypted cloud data by considering the weights of search keywords [32]. Qian et al. proposed a scheme named *VoiceMask* [33] adding an intermediary process entity between users and the cloud to anonymize speech data before sending it to the cloud for speech recognition. Our anonymizer employs a privacy-preserving matching scheme, commonly used to check the similarity of two profiles without disclosing them [34, 35]. Furthermore, our scheme is designed based on a new similarity calculation considering the unique VAS usage patterns.

Another problem of VAS devices is voice authentication [36, 37]. The popular devices such as Alexa, Siri, and Google home do not have mechanism to assure the voice commands originated from specific user account binding with VAS devices. Many researchers have studied voice authentication: Feng et al. [38] proposed continuous authentication as a mechanism to guarantee that the voice assistant system executes only the commands that originate from the voice of the owner. It uses a wearable security token to collect the body-surface vibrations of the speaker and continuously correlate the vibration signals to the voice signal for speaker identification. Chandrasekaran et al. [39] proposed a method to allow VAS devices to work in two different modes: a privacy-preserving mode and a normal mode. In a privacy-preserving mode, Chandrasekaran exploited extended devices or software to intervene the voice recording process in the VAS. Through out this intervention, VAS devices cannot record and upload sensitive voice data to the server. Our research does not focus on limiting the voices data to be uploaded. However, for uploaded voice data, we aim to disable the linkage of the uploaded voice data and the source of the voice data for anonymizable VAS services.

8 Conclusion

In this paper, we proposed a novel anonymizer on the voice assistant devices for protecting users' voice data from being linked to their accounts by the service provider. The anonymizer aims to mix the queries from multiple VAS users' devices, hiding the source of queries and hiding the relay's real queries. To achieve effective anonymity, the anonymizer is equipped with a proposed privacy-preserving pattern matching scheme, which is run with the help from a semi-trusted server and is used to find the most effective relay for the requester based on their pattern similarity. To enhance the effectiveness of the anonymity protection, we proposed anonymity evaluation modules, which allow both requester and relay to evaluate the real-time query generated at requester. The matching scheme will be run periodically or when the rejecting rate at relay is non-tolerable. We evaluated the privacy and efficiency of the proposed matching scheme in realistic network settings and system parameters. The matching scheme has been shown to fully protect the patterns from disclosure, and it is performed effectively and efficiently at the VAS devices in realistic network conditions. We tested the anonymity evaluation modules and found that the modules at both requester and relay largely reduces the communication overhead and enhances the anonymity protection. Our future work includes a full implementation of the VAS anonymizer and an evaluation of VAS user experiences. We will also

enhance our anonymizer by generating fake queries that are semantically and syntactically similar to the real queries, as well as design a multi-hop anonymizer scheme to improve anonymity.

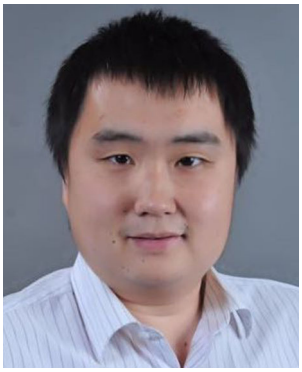
Acknowledgements This research project is supported by the US National Science Foundation award number CNS-1618893 and the National Institutes of Health National Institute on Aging award number R01AG067416. The views and conclusions in this document are those of the authors and may not necessarily represent the official policies of NSF and NIH.

References

1. Kinsella, B, Juniper estimates 3.25 billion voice assistants are in use today, google has about 30% of them. https://voicebot.ai/2019/02/14/
2. Smith, S, Digital voice assistants in use to triple to 8 billion by 2023, driven by smart home devices. www.juniperresearch.com
3. Alexa goes to college (2018) Echo dots move into dorms on campus. https://www.usatoday.com/story/money/2018/09/06/college-students-echo-dots-dorm-rooms/1087251002/, [accessed 10-August-2020]
4. Welch (2018) Amazon made a special version of alexa for hotels that put echo speakers in their rooms. https://www.theverge.com/2018/6/19/17476688/amazon-alexa-forhospitality-announced-hotels-echo/
5. Voicebot.ai (2020) Nearly 90 million u.s. adults have smart speakers, adoption now exceeds one-third of consumers. https://voicebot.ai/2020/04/28/nearly-90-million-u-s-adults-have-smart-speakers-adoption-now-exceeds-one-third-of-consumers/. [Online; accessed 10-May-2020]
6. Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492
7. Schalkwyk, J, An all-neural on-device speech recognizer. https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html
8. Coucke A, Saade A, Ball A et al (2018) Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190
9. Zhu Y, Li X (2020) Privacy-preserving k-means clustering with local synchronization in peer-to-peer networks. Peer-to-Peer Networking and Applications, pp 1–13
10. Wang Z, Song M, Zhang Z, Song Y, Wang Q, Qi H (2018) Beyond inferring class representatives:, User-level privacy leakage from federated learning. arXiv preprint arXiv:1812.00535
11. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) 10(2):1–19
12. Differential Privacy Team at Apple: Learning with privacy at scale. https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html
13. Liu X, Li H, Xu G, Lu R, He M (2020) Adaptive privacy-preserving federated learning Peer-to-Peer Networking and Applications
14. Dingledine R, Mathewson N, Syverson P (2004) Tor: the second-generation onion router. Tech. rep., Naval Research Lab Washington DC
15. Rainie L, Kiesler S, Kang R, Madden M, Duggan M, Brown S, Dabbish L (2013) Anonymity, privacy, and security online. Pew Research Center, p 5
16. My activity at google. https://myactivity.google.com/myactivity
17. Smart speaker consumer adoption report. https://voicebot.ai/wp-content/uploads/2018/10/voicebot-smart-speaker-consumer-adoption -report.pdf
18. Lopatovska I, Rink K, Knight I, Raines K, Cosenza K, Williams H, Sorsche P, Hirsch D, Li Q, Martinez A (2019) Talk to me: Exploring user interactions with the amazon alexa. J Librariansh Inf Sci 51(4):984–997
19. Kim C, Misra A, Chin K, Hughes T, Narayanan A, Sainath T, Bacchiani M (2017) Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home. In: Interspeech, pp. 379–383
20. Diffie W, Hellman M (1976) New directions in cryptography. IEEE transactions on Information Theory 22(6):644–654
21. Li H, Lu R, Mahmoud MM (2020) Security and privacy of machine learning assisted p2p networks. Peer-to-peer networking and applications, pp 1–3
22. Avs device sdk. https://github.com/alexa/avs-device-sdk (2020). [accessed 20-October-2020]
23. Amazon web services. https://aws.amazon.com/ (2020). [accessed 10-August-2020]
24. Ubuntu wiki. https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes (2020). [accessed 10-August-2020]
25. Tp-link 450mbps wireless n router. https://www.tp-link.com/us/home-networking/wifi-router/tl-wr940n/ (2020). [accessed 10-August-2020]
26. Pathak MA, Raj B, Rane SD, Smaragdis P (2013) Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise. IEEE signal processing magazine 30(2):62–74
27. Gao C, Chandrasekaran V, Fawaz K, Banerjee S (2018) Traversing the quagmire that is privacy in your smart home. In: Proceedings of the 2018 Workshop on IoT Security and Privacy, pp. 22–28. ACM
28. Hadian M, Altuwaiyan T, Liang X, Li W (2019) Privacy-preserving voice-based search over mhealth data. Smart Health 12:24–34
29. Qian J, Du H, Hou J, Chen L, Jung T, Li XY, Wang Y, Deng Y (2017) Voicemask:, Anonymize and sanitize voice input on mobile devices. arXiv preprint arXiv:1711.11460
30. Qian J, Han F, Hou J, Zhang C, Wang Y, Li XY (2018) Towards privacy-preserving speech data publishing. In: IEEE INFOCOM, pp. 1079–1087
31. Glackin C, Chollet G, Dugan N, Cannings N, Wall J, Tahir S, Ray IG, Rajarajan M (2017) Privacy preserving encrypted phonetic search of speech data. In: IEEE ICASSP, pp. 6414–6418
32. Li H, Yang Y, Luan TH, Liang X, Zhou L, Shen XS (2016) Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Transactions on Dependable and Secure Computing 13(3):312–325
33. Qian J, Du H, Hou J, Chen L, Jung T, Li XY (2018) Hidebehind: Enjoy voice input with voiceprint unclonability and anonymity. In: Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, pp. 82–94. ACM
34. Liang X, Li X, Zhang K, Lu R, Lin X, Shen XS (2013) Fully anonymous profile matching in mobile social networks. IEEE Journal on Selected Areas in Communications 31(9):641–655
35. Rabieh K, Mahmoud M, Siraj A, Misic J (2015) Efficient privacy-preserving chatting scheme with degree of interest verification for vehicular social networks. In: IEEE GLOBECOM, pp. 1–6
36. Meng Y, Wang Z, Zhang W, Wu P, Zhu H, Liang X, Liu Y (2018) Wivo: Enhancing the security of voice control system via wireless signal in iot environment. In: Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 81–90. ACM

37. Yuan X, Chen Y, Zhao Y, Long Y, Liu X, Chen K, Zhang S, Huang H, Wang X, Gunter CA (2018) Commandersong: a systematic approach for practical adversarial voice recognition. In: 27Th USENIX security symposium (USENIX security 18), pp. 49–64
38. Feng H, Fawaz K, Shin KG (2017) Continuous authentication for voice assistants. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, pp. 343–355. ACM
39. Chandrasekaran V, Fawaz K, Mutlu B, Banerjee S (2018) Characterizing privacy perceptions of voice assistants: A technology probe study. arXiv preprint arXiv:1812.00263

Bang Tran received his M.S degree of Computer Science from Ho Chi Minh City, Vietnam National University in 2010. He had worked as a lecturer for University of Transport and Communications, Vietnam for fifteen years. Bang Tran is currently pursuing his Ph.D degree at Computer Science Department of University of Massachusetts, Boston (UMB), USA. His research interests are Security and Privacy for IoT systems, big data privacy and applied privacy.



Xiaohui Liang received his Ph.D degree in Electrical and Computer Engineering from University of Waterloo, Canada, in 2013. He is an Assistant Professor of the Department of Computer Science at University of Massachusetts, Boston (UMB), USA where he leads the Mobile Computing and Privacy (MobCP) Lab. His research interests are Mobile Healthcare, Internet of Things, Wearable Computing, and Security and Privacy for Communication and Networking Systems.