Computer Networks 133 (2018) 59–72

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# Privacy-preserving sparse representation classification in cloud-enabled mobile applications

Yiran Shen[a], Chengwen Luo[b], Dan Yin[a,*], Hongkai Wen[c], Rus Daniela[d], Wen Hu[e]

[a] College of Computer Science and Technology, Harbin Engineering University, China
[b] College of Computer Science and Software Engineering, Shenzhen University, China
[c] Department of Computer Science, University of Warwick, UK
[d] Computer Science and Artificial Intelligence Laboratory, MIT, USA
[e] School of Computer Science and Engineering, UNSW Australia, Australia

ARTICLE INFO

*Article history:*
Received 25 June 2017
Revised 13 November 2017
Accepted 18 January 2018
Available online 30 January 2018

*Keywords:*
Sparse representation classification
Mobile computing
Privacy preserving

ABSTRACT

Mobile devices are now pervasive to provide prolific services to the users meanwhile collect the information derived from the activities of the individuals with the onboard sensors. Classification and authentication are popular provided services of many mobile applications which have high probability to involve the sensitive information of the users. In this paper, we propose a new cloud-enabled and Privacy-preserving sparse representation classification ($\mathcal{P}^2$-SRC) system to protect the privacy of both the "data contributors" and "application users" when cloud server is *untrusted*. Different from the state-of-the-art approaches which only consider the attacks on data values, our proposed system, $\mathcal{P}^2$*-SRC*, addresses multiple types of privacy attacks including *Content Privacy Attacks, Source Privacy Attacks* and *Label Privacy Attacks*. As a result, besides the data values, in $\mathcal{P}^2$-SRC, the identities and activities of the users are also protected. According to our evaluations on two different classification applications (face recognition and activity recognition), $\mathcal{P}^2$-SRC achieves almost the same classification accuracy compared with traditional SRC approach which indicates the security add-ons do not affect the accuracy of the SRC classifier. We also demonstrate that it outperforms the most related work, Pickle, significantly on recognition accuracy and privacy protections. Meanwhile the implementation of $\mathcal{P}^2$-SRC in a face recognition application on smartphones demonstrates that $\mathcal{P}^2$-SRC based authentication system accounts for only 0.000041% of the total energy supply of a normal smartphone and the average responding time is around 1.1 s for each recognition request.

© 2018 Elsevier B.V. All rights reserved.

## 1. Introduction

Cloud-enabled mobile applications (CMAs) are booming with the pervasive availability of smart mobile devices (e.g., smartphones, tablets, wearable devices), high speed networks (e.g., Wifi and 4G) and high performance cloud services [19,28]. As CMAs process large amount of crowdsourced data in centralised manner, most of the CMAs utilise the resources of cloud servers to store overwhelming amount of collected data and undertake computationally intensive tasks.

Signals classification is popular in CMAs and it has been studied in the literature for decades [24]. It is the basis of the authentication [9], medical diagnosis [5] and environment awareness systems [25]. Crowdsourcing benefits the classification system especially for learning the classification model (i.e., the classifier). For example, to develop a classification model, the application publisher needs to collect sufficient training samples from certain group of people where crowdsourcing saves the efforts on the data collection. In this paper we define two risky groups of subjects whose privacy can be attacked in the CMAs: the *Data Contributors* and *Application Users*. Data contributors are recruited by application publishers to upload their sensor data of their mobile devices to the cloud for building the training set. The application users make use of the built classification models for recognition/classification.

Innovate CMAs have created many possibilities [4,26,27,30]; however, security issues arise when cloud server is not trusted [46,55]. The uploaded sensor data of mobile devices may contain personal information or be used to infer individuals' private information. To prevent the privacy leakage, many researchers have proposed new privacy-preserving methods [12,25,53]. For instance, Pickle was proposed by Liu et al. [25] to provide certain

[*] Corresponding Author.
*E-mail addresses:* shenyiran@hrbeu.edu.cn (Y. Shen), yindan@hrbeu.edu.cn (D. Yin).

https://doi.org/10.1016/j.comnet.2018.01.035
1389-1286/© 2018 Elsevier B.V. All rights reserved.

privacy protection in the classification system when learning the classifier from the encrypted crowdsourced training set. However, Pickle only considered the *content privacy* attacks (sensor values) but ignored the possible *label privacy* and *source privacy* attacks (see definitions next paragraph) though they admitted labels may leak important information.

**Three types of attacks** In this paper we introduce three types of privacy attacks, i.e., *Content Privacy Attacks, Source Privacy Attacks* and *Label Privacy Attacks*.

- Content privacy attacks are the attacks on drawing actual values of the sensor data of mobile devices.
- Source privacy attacks aim to find the sources or related identities where the sensor data is derived;
- Label privacy attacks are the attacks on obtaining class labels of the training set which allows the adversaries to interpret the classification results and users' activities.

**The sources** of the sensor data are sensitive information. For example, in a cloud-enabled authentication system, some of the computational burden is shifted to the cloud. The authentication decision is made according to the results computed on the cloud. If the sources are not protected and the cloud is compromised, the adversaries may collude with the cloud server to deduce the possible results from the historical observations and send back the "fake" results to the mobile devices to help the adversaries to break into the authentication systems. **The class labels** can be used to interpret the classification activities. In label privacy attacks, if the class labels are not protected, the adversary is able to obtain the specific physical meaning of the class labels. Some private information of the users or data contributors can be drawn from the classification results. For examples, the attackers are able to deduce the health status of the patients in the medical diagnosis system or track the daily activities of the individuals in the activity recognition system. However, health status and daily activities are extremely sensitive and most of the users are not willing to disclose them to the third party or to the public. Meanwhile, They have substantial commercial values so that are very likely to be targeted by the attackers.

To solve the above mentioned three types of privacy issues meanwhile providing reliable classification services, we propose a new Privacy-Preserving and cloud-enabled classification system, $\mathcal{P}^2$-SRC, based on Sparse Representation Classification (SRC), for CMAs. SRC is an emerging classification method and it has demonstrated superior performance on recognition accuracy compared with other traditional classification methods and is successfully used in face recognition [39,48], wildlife sound recognition [45] and activities classification [44]. Besides the accurate classification services, $\mathcal{P}^2$-SRC uses random projection matrices to compress sensor data meanwhile protect the data content. Then a Tor-like network is incorporated in the SRC-based classification framework to protect users from label and source privacy attacks. The contributions of this paper are as follows:

- Overall, we propose a new privacy-preserving and cloud-enabled classification framework, $\mathcal{P}^2$-SRC. As our evaluations on different classification applications, its recognition accuracy is almost the same to the traditional SRC method which indicates the privacy add-ons do not deteriorate the recognition accuracy of SRC.
- $\mathcal{P}^2$-SRC addresses different types of privacy attacks including content privacy attacks, source privacy attacks and label privacy attacks. To the best of our knowledge, it is the first privacy-preserving classification system that can address all of the three types of privacy attacks. Meanwhile it achieves significant improvement on accuracy-privacy trade-off according to our evaluations on two classification applications.
- We conduct two user studies to demonstrate, in intuitive approach, 1) $\mathcal{P}^2$- SRC provides reliable protections for users' data values, and 2) most of the users concern more about the protections of label and source information than sensor data values.
- At last, we implement a face recognition system based on $\mathcal{P}^2$-SRC on smartphones. The results show that the system cost of $\mathcal{P}^2$-SRC is negligible, and it provides real-time responses.

The organization of the rest of this paper is as follows. We first review the related literature in Section 2. Then we provide a brief introduction of SRC in Section 3. In Section 4, we discuss the system architecture, present two application examples and provide privacy analyses of $\mathcal{P}^2$-SRC. Section 5 evaluates the performance of two classification applications of $\mathcal{P}^2$-SRC with two publicly available datasets. Section 6 evaluates the system cost of the implementation of $\mathcal{P}^2$-SRC face recognition application on mobile devices. Finally we conclude the whole paper in Section 7.

## 2. Related work

In this section, we will give a literature review on the state of the arts in relevant research area. As our proposed system aims to protect the privacy of the users in mobile classification applications and the classification engine is the Sparse Representation Classifier (SRC), we discuss the recent advances in privacy protection for CMAs and applications of SRC.

### 2.1. Privacy protections for mobile applications

Mobile app development is a hotspot and the security of the mobile devices has become one of the most recent major concerns in mobile system research community [2,21,31,35]. For examples, Herberst et al. [17] designed privacy capsule to avoid the private information leakage via the untrusted third party mobile apps. While Zhu et al. [56] studied the private information leakage in code level; they addressed the 'module-level attacks' of the mobile app to prevent the third-party code stealing the private information on the COTS mobile devices. To protect the mobile devices from illegal usage meanwhile provide non-intrusive authentications, touch input implicit authentication (touch IA) was proposed and studied on different mobile devices; however, as the evaluations by Khan [22], touch IA was easy to be mimicked and not suitable from a security standpoint.

Private information can be vulnerable when sharing or uploading data via wireless channel. Xu et al. [49] proposed Walkie–talkie to generate encryption key based on gait. It enabled the mobile devices get paired automatically and prevents the devices from eavesdropping radio communications. Chakraborty et al. [7] proposed Ipshield a new context-aware privacy protection scheme to estimate the risks of sharing data in the cloud-enabled mobile applications. It provides the user with a list of the inferences that can be drawn from the data so the user can be aware of potential risks when sharing the data. To deal with the problem of reliability of the sensory data provided by the data contributors, Miao et al. [29] proposed a cloud-enabled privacy-preserving truth discovery framework which solved the problem of private information protection existed in the previous truth discovery approaches. Poolview [15] studied the problem of reconstruction attack and proposed a synthesis model which introduced correlated noise perturbation to protect the privacy of data sharing. Liu et al. proposed Pickle [25] which is the most related work to $\mathcal{P}^2$-SRC. Pickle enabled privacy-preserving collaborative learning for SVM using a linear regression based approach. However, it only considered the problem of content privacy attack. Other solutions [18,33] allowed users to control their resources and created shadow to prevent the untrusted applications to access the resources on their mobile devices. The most advantages of $\mathcal{P}^2$-SRC over the existing works are the miscellaneous privacy protections it provides including content privacy as well as label privacy and source privacy.

### 2.2. Applications of SRC

SRC has been popularly used in improving the performance of applications on embedded systems including face recognition [39,40], cognitive assistance [50,51], sound classification of wildlife animals [45], activity recognition [44,52] and video tracking [37,38]. Random projection matrices are used in [45,48] to reduce the dimensionality of the problems while preserving the accuracy of classification. Shen et al. [39] studied the problems on how to optimize the projection matrix to improve the classification accuracy with the knowledge of the training dictionary.

## 3. Sparse representation classification

Huang et al. [20] formulate the signals classification as a sparse representation problem computed via $\ell_1$ optimization (termed as SRC). The formulation uses a random projection matrix for dimensionality reduction. The steps of SRC are:

### 3.1. Dictionary building

To model signals classification as a sparse representation problem, one needs to first build a dictionary $\mathcal{D}$. We assume there are $K$ classes and $T$ training samples per class. Each training sample is an $N$-dimensional column vector. We then assemble the training samples of the $i$th subject in a $N \times T$ sub-dictionary $D_i$. Then a $N \times KT$ dictionary $\mathcal{D} = [D_1, D_2, \dots, D_K]$ is formed from the $K$ classes.

### 3.2. Sparse representation

Let $y$ denote a test vector, its representation under the dictionary $\mathcal{D}$ is obtained by solving the following linear equation with the knowledge of $y$ and $D$:

$$y = \mathcal{D}\theta \tag{1}$$

where the unknown vector $\theta$ contains $n = KT$ unknowns which is equal to the number of columns in $\mathcal{D}$. If the test signal $y$ belongs to the $k$th class, then *ideally* $y$ is within the space spanned by the $T$ vectors in $D_k$ class and independent of the other classes. If the ideal condition holds, then the representation vector $\theta$ for $y$ has the form:

$$\theta = [0, 0, \dots, \alpha_{k,1}, \alpha_{k,2}, ., \alpha_{k,T}, \dots, 0, 0, ., 0]^T \tag{2}$$

where $\cdot^T$ denotes the matrix transpose, and the non-zero elements appear only in those positions related to the $k$th class in $\mathcal{D}$. If the number of classes $K$ is large, then $\theta$ is a *sparse* vector if the ideal condition holds.

### 3.3. Random projections

In the applications using SRC, the dimension $N$ of the signal vector is huge, solving Eq. (1) can be computationally expensive for mobile devices. A random projection matrix can be applied to improve the computational efficiency while preserving recognition accuracy. The random projection matrix $\Phi$ in most of the applications on embedded or mobile system [39,45] is generated from a Gaussian distribution with zero mean and unit variance and does not consider the prior knowledge of the dictionary. Incorporating an $m \times p$ Gaussian matrix $\Phi$ in Eq. (1), we have

$$\Phi y = \Phi\mathcal{D}\theta. \tag{3}$$

where $m \ll n$ makes the systems of linear equations underdetermined. Since we are looking for a sparse representation $\theta$, we aim to solve the following $\ell_0$ optimization problem

$$\hat{\theta} = \arg\min \|\theta\|_0 \quad \text{subject to } \Phi y = \Phi\mathcal{D}\theta \tag{4}$$

where $\hat{\theta}$ is the sparse representation of $y$ under dictionary $\mathcal{D}$ and $\|\cdot\|_0$ represents the $\ell_0$ norm, which counts the number of non-zero coefficients in $\hat{\theta}$. The optimization problem (4) is *NP-hard* [23], which means no known algorithms can solve the problem within polynomial time.

Inspired by the recent theory of CS, the solution of $\ell_0$ optimization in Eq. (4) can be well approximated by the following $\ell_1$ optimization problem,

$$\theta_{\text{opt}} = \arg\min \|\theta\|_1 \quad \text{subject to } \|\Phi y - \Phi\mathcal{D}\theta\|_2 < \epsilon \tag{5}$$

where $\epsilon$ is a small positive value used to account for noise. The solution $\theta_{\text{opt}}$ from the $\ell_1$ optimization is used in the following classification procedure.

### 3.4. Minimal residual

After obtaining the coefficient vector $\theta_{\text{opt}}$, we can determine the class of the test vector $y$ by using residuals. The residual for class $i$ is:

$$r_i = \|y - \mathcal{D}_i\theta_{\text{opt}}^{(i)}\|_2 \tag{6}$$

where $\theta_{\text{opt}}^{(i)}$ is a $T$-dimensional vector containing the $T$ elements in $\theta_{\text{opt}}$ related to class $i$. Then the final classification is determined by

$$\hat{i} = \arg\min_{i=1,2,\dots K} r_i, \tag{7}$$

i.e., the class having the minimal residual among all classes.

## 4. Privacy-preserving SRC

### 4.1. Threat model

In this paper, we consider an *honest-but-curious* cloud server. It follows the protocol to provide training set building and classification services but is curious about users' private information such as data content, data sources, data labels, etc. In addition, the peer users are also not trustworthy: they might collude with the cloud server in order to obtain the private information of the legitimate users. A similar threat model, which only considers the data content privacy, is widely adopted in recent cloud-enabled systems [6,10,25], and our goal in this work is to protect the content, source and label privacies of both data contributors and application users.

### 4.2. System overview

The system architecture can be vastly divided into two main parts. The first is the privacy-preserving data collection where sensor data samples are collected from data contributors to form the training set on cloud server. The second is the cloud-enabled and privacy-preserving classification where the application users send their encrypted test samples to the cloud server to compute the sparse representation vectors then using the returned sparse representation vectors to determine the final classification results (see Fig. 1).

Specifically, in the privacy-preserving data collection stage, the application publishers first advertise their requirements for recruiting data contributors, the random projection matrix (or called

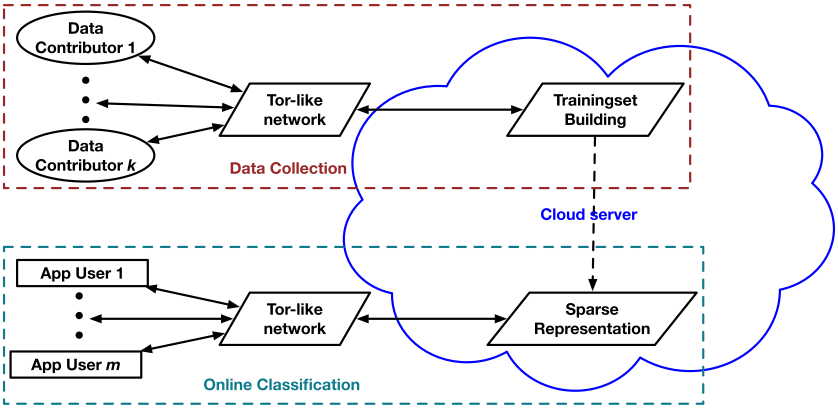


**Fig. 1.** System architecture of $\mathcal{P}^2$-SRC.

compressed key with respect to its privacy protection purpose), detailed privacy risks and incentives to be provided. Then the recruited data contributors collect, compress and transmit the required sensor data to the cloud server via the anonymous communication channel. With the anonymous communication protocol, the cloud server is unknown who contributes the training samples and their physical meanings. Finally the training set is formed by the encrypted data samples uploaded from different data contributors. It is worth noting that all the data contributors use **the same** compressed key so that the compressed key is very likely to be disclosed.

As SRC-based classification systems do not involve model learning, the system is ready for the classification tasks once the training set is built. Similar to the training phase, the application users collect, compress (using the same compressed key) and transmit the test data to the cloud server via anonymous communication channel. The sparse representation of the test data is computed on the cloud server. The resultant sparse representation vector is returned back via anonymous communication channel. At last, the final classification decision is obtained by computing the minimal residual with the knowledge of the class labels on the local mobile devices.

### 4.3. Privacy-preserving data collection

In privacy-preserving data collection, protection of content is achieved via applying compressed key while sources and labels are protected by introducing anonymous communication channel in the SRC-based classification framework.

#### 4.3.1. Random projections for content protection

Before data contributors upload their sensor data samples to the cloud server, the compressed key (or random projection matrix) obtained from the application publisher is first applied to protect the content of data. For example, one of the data contributor $i$ collects a sensor data vector $d_{i,t} \in \mathbb{R}^N$ at time $t$. Then the data vector is encrypted and compressed using the compressed key $K \in \mathbb{R}^{M \times N}$ $(M \ll N)$,

$$a_{i,t} = K \cdot d_{i,t} \tag{8}$$

We call the random projection matrix as *Compressed Key* because it both compresses and encrypts the original data vector. Random projection matrix provides a computational guarantee of secrecy [34] so that reconstruction with a wrong projection matrix will produce incorrect results. As the number of elements in the matrix is huge and they are randomly generated from Gaussian distribution, the brute force attack is not feasible. Besides, in $P^2$-SRC, adversaries cannot reconstruct the original sensor data even if the compressed key is disclosed. This will be discussed further in Section 4.6 and evaluated in Sections 5.2.3 and 5.3.3.

After the data vector is compressed and encrypted, it will be uploaded to the cloud server via anonymous communication protocol to form the training set.

#### 4.3.2. Anonymous communication channel for source and label protections

In a cloud-enabled approach, the encrypted data vector $a_{i,\ t}$ needs to be uploaded to the cloud server as a training sample. Such approach is termed as *collaborative learning* [30] in the literature and is widely adopted by researchers to build high quality classification models to support novel mobile applications. However, although each uploaded data vector is encrypted by the compressed key, the equal important sensitive information, i.e., the sources (identities of the data contributors) and class labels, are revealed during the uploading process; (users concerns more about the protection of labels and sources according to the user studies in Section 5.4). In previous work [7,25] the source and class label of each uploaded feature vector is known to the cloud, the correspondence between the identity and the uploaded training data may lead to potential tracking attacks. For example, if the data vector contains the encrypted voice features of the data contributors, even though the cloud server cannot decrypt the content efficiently due to the usage of compressed key, during classification process the cloud server is still able to tell one encrypted input voice segment belongs to which data contributor. Therefore, exposing the sources during dictionary building potentially allows the cloud to track data contributors by voice features in future cloud services. Similarly, exposing the label of each uploaded data vector also poses significant privacy-leakage threats to data contributors. For example, if the subject classes are different types of diseases, the data contributors and application users might not want the cloud to gain the knowledge about the label of each uploaded data vectors. Based on the above concerns, in this work, we consider a significantly more stricter privacy-preserving scheme to protect the content, source, and label of each uploaded data vector.

*4.3.2.1. Anonymous communication channel.* To protect the identities of communication entities, many researches have been done to design anonymous communication channels. One popular approach is through the use of Tor network [13]. The Tor network relies on the intermediate message relays called *Tor proxies* and the *onion routing* which is implemented by encryption in the application layer of a communication protocol stack to achieve anonymous communication. However, researchers have reported several shortfalls in the Tor network [32]: (i) when constructing the routing path, relays with low bandwidth have higher probability of being selected; (ii) Tor servers are scattered around the world, resulting a large RTT during communication; (iii) lack of active servers.

Since most CMAs assume many participating mobile devices, a less complicated, more light-weight, more flexible and dynamic solution is possible by exploiting the pervasive communication capability of the participating mobile devices. Therefore, we implement a Tor-like network on mobile devices in $\mathcal{P}^2$-SRC to achieve hard-to-trace communication through a chain of proxy servers, i.e., other peer mobile devices in $\mathcal{P}^2$-SRC. In the Tor-like network, all the participating mobile devices also serve as proxies and the cloud maintains and distributes the list of the active mobile nodes. When constructing the Tor-like network, the end mobile device randomly selects multiple available peers from the list received from the cloud as its proxies and sends its encrypted message through the network. Each proxy server takes the input messages from the previous proxy, decrypts the outermost layer of the onion encryption [8] of the address trace to reveal the next proxy's address then sends the remaining messages to the next proxy until the messages reach the final destination. As a result, the final recipient cannot trace the source of any message due to the intermediate proxies. In our Tor-like network, onion encryption only applies on the address trace as the original message has been protected by compressed key. Therefore, the computation and communication cost can be reduced. As shown in our evaluation results, the Tor-like network achieves low latency while also incurs small energy consumptions on mobile devices. The detailed description of the Tor-like network is as below:

This Tor-like network is based on the architecture of mix network. The public key cryptosystem is adopted in the network. For a Tor-like network consists of $n_m$ proxies, a pair of public/private key $(K_1, K_1^{-1}), (K_2, K_2^{-1}), \ldots, (K_{n_m}, K_{n_m}^{-1})$ are generated by each proxy such that for $i = 1, \ldots, n_m$ and arbitrary message $M$ ($M$ is the projection vector in $\mathcal{P}^2$-SRC), $K_i^{-1}(K_i(M)) = M$. In $\mathcal{P}^2$-SRC, as the message has been encrypted by the compressed key, the implementation of mix network is modified and is only applied to protect the uploading trace consisting of the proxies and cloud's addresses.

To send a message to the cloud server through only one proxy, the data contributor uses the public key of the server $K_s$ and the public key of the proxy $K_1$ to encrypt the message $M$ and server's address $A_s$ successively:

$$K_1(R_1, K_s(R_s, M), A_s))).$$

The proxy receives the message, decrypts it using the private key $K^{-1}$, throws away the random string $R_1$ and sends the remainder $K_s(R_s, M)$ to the cloud server using the address $A_s$. The server decrypts the message using its private key $K_s^{-1}$, throws away the random string $R_s$ and obtains the original message $M$. The random strings are applied to prevent the adversaries to infer the data vector with the available public keys which is also known as the collision attack [36]. Since the correspondence between the received message on the server side and the input of the proxy is eliminated by the proxy, the source of the message $M$ is hidden from the cloud server.

Considering the fact that one proxy may be not *trustworthy*, a *cascade*, or a series of proxies are applied so that any single proxy in the network can guarantee the anonymization of the source. To use a cascade, the data contributor chooses a sequence of $n_m$ proxies with addresses $A_1, A_2, \ldots ,A_{n_m}$, and encrypts the addresses of the trace to the cloud server as:

$$K_{n_m}(R_{n_m}, K_{n_m-1}(R_{n_m-1}, \ldots,$$
$$K_2(R_2, K_1(R_1, A_s), A_1) \ldots, ), A_{n_m-1}).$$

where $K_s$ is the address of the server and $K_1, K_2 \ldots, K_{n_m}$ are the public keys of the proxies.

The first proxy receives the above message and the encrypted data vector, decrypts one layer of the encrypted message using its private keys, and obtains the message:

$$R_{n_m}, K_{n_m-1}(R_{n_m-1}, \ldots,$$
$$K_2(R_2, K_1(R_1, A_s), A_1) \ldots, ), A_{n_m-1}.$$

The proxy then throws away the random string $R_{n_m}$ and sends the remainder and the encrypted data vector to the next proxy as the decrypted address $A_{n_m-1}$ (all the revealed addresses are also discarded). The random strings are applied to prevent the adversaries to infer the data vector with the available public keys which is also known as the collision attack [36]. In this way, the encrypted data vector finally reaches the cloud server after passing through $n_m$ proxies. Each proxy only knows the addresses of its one-hop neighbours. As a result, as long as one proxy in the cascade is trustworthy, the correspondence between the input data vector and the final received data vector on the server can be eliminated and hence the sources of the data vector is protected.

*4.3.2.2. Practical Tor-like network protocol.* One potential problem of the Tor-like network is the system scalability. The length of the message and communication delay grows with the number of proxies selected. Therefore, the scheme is not scalable and becomes impractical if the number of selected proxies is large during the dictionary building. To tackle the scalability problem, each data contributor randomly choose fixed number ($n_m$) of other mobile devices participating in the same application as the proxies and decides the trace to the cloud server locally. As $n_m$ is fixed, the computational cost for uploading each message does not change as the number of mobile devices increases in the system. As the privacy analysis in Section 4.6.2, when $n_m = 5$, it achieves high probability (0.97%) of source protection even if half of the mobile devices is possible to collude with the cloud server. At last, to address the label privacy attack, during data uploading, the data contributors only upload the encrypted data vectors and hide the label of each data vector. As a result, $\mathcal{P}^2$-SRC hides both the source and label of each data vector.

### 4.3.3. Training set multicast

After the training set is built from the collected training samples on the cloud server, the labels of the training set are still kept anonymous to all parties as the data contributors upload sensor data without label information. The training set is then multicast to all the data contributors so that each contributor is able to confirm the column indices of its submitted samples in the training set (i.e., the class labels used in the future classification applications) by directly comparing the column vectors of the training set with the data vectors they uploaded to the cloud server previously. As the labels are generated locally, they are protected from the cloud server and other users. The training set and generated class labels are stored locally at the data contributors (e.g., end devices in the authentication application) or uploaded to a trusted third party (e.g., a doctor in a medical diagnosis application) depending on different classification scenarios. Training set multicast does not introduce further information leakage because the training set is encrypted by compressed key; the class labels and sources information are not included in the training set.

It is worth noting that SRC-based classification method **does not involve model learning stage**. However, learning a classification model is an essential component for most of the traditional classification methods, e.g., SVM, and the available labels information is required. It is known that K-Nearest Neighborhood (K-NN) does not involve model learning either, however, as the evaluations in Section 5, its classification accuracy is significantly lower than SRC. Therefore, we choose SRC as the fundamental building block for our system.

### 4.4. Cloud-enabled and privacy-preserving classification

After the privacy-preserving data collection, cloud server has obtained the training set $\mathcal{A}$ which consists of the compressed and encrypted data vectors as its columns. Due to the usage of Tor-like network, the server does not have the information about the source and label of each data vector. Different from the training set in traditional SRC application where the data vectors with the same class labels (or sources) are grouped together, the cloud side of $\mathcal{P}^2$-SRC randomly places the data vectors regardless of their class labels or sources as they are unknown to cloud server.

In the classification stage, the application user $i$ first obtains a data vector $x_{i,\ t}$ and compresses it with the compressed key $\mathcal{K}$. It can be expressed as,

$$y_{i,t} = \mathcal{K} \cdot x_{i,t} \tag{9}$$

Then $y_{i,\ t}$ is uploaded to the cloud via the Tor-like network for computing its sparse representation using $\ell_1$ optimisation.

**Returning address encryption** Different from the data collection stage, the application user should also upload the encrypted returning trace to the cloud server to guide the sparse representation vector back to the application user. The returning trace is successively encrypted in a reverse manner of the encryption of uploading address.

$$K_s(K_1(\ldots, K_{n_m}(A_u, R_u)\ldots, A_2, R_1)A_1, R_1),$$

where $A_u$ is the address of the application user. The most outer layer of the encryption is the public key of the server, therefore the intermediate proxies cannot start decrypting the layers of the union encryption of the returning address trace until the encrypted data reaches the cloud server.

#### 4.4.1. Labelless cloud-enabled sparse representation

Computing sparse representation accounts for most of the resource consumptions of the SRC-based classification on mobile devices [39,45]. Considering mobile devices are resource constrained, we shift the computational burden of sparse representation to the cloud server.

When the cloud server receives $y_{i,\ t}$ from some unknown application user, the sparse representation vector will be computed by solving $\ell_1$ optimisation,

$$\hat{\theta}_{i,t} = \arg\min\|\theta_{i,t}\|_1 \quad \text{subject to } \|y_{i,t} - \mathcal{A}\theta_{i,t}\|_2 < \epsilon \tag{10}$$

where $\hat{\theta}_{i,t}$ is the estimated sparse representation of $y_{i,\ t}$. under training set $\mathcal{A}$. According to the form of Eq. (10), computing sparse representation does not involve class labels. On the contrary, the most popular used classifier, SVM (used in Pickle), needs the class labels to train the classification models.

#### 4.4.2. In-situ residual computation on mobile devices

After computing $\ell_1$ optimisation on the cloud server, the sparse representation vector $\hat{\theta}_{i,t}$ is returned back to the application user $i$ via the Tor-like network by decrypting the encrypted returning trace layer by layer.

The application user receives the sparse representation vector $\hat{\theta}_{i,j}$ and computes the residuals on the local mobile devices to determine the class of the test vector $y_{i,\ t}$. To further speed up the classification and reduce the energy consumption on the mobile devices. We adopt the *Compressed Residual* proposed by Shen et al. [39]. The compressed residual of class $j$ is expressed as,

$$r_{i,t} = \|y_{i,t} - \mathcal{A}_j\theta_{i,t}^{(j)}\|_2 \tag{11}$$

where $\theta_{i,t}^{(j)}$ contains the $T$ elements in $\theta_{i,\ t}$ related to class $j$. Then the final classification result is determined by

$$\hat{j} = \arg\min_{j=1,2,\ldots M} r_{i,t}, \tag{12}$$

where $M$ is total number of classes. The class produces the minimal residual is the final classification result.

### 4.5. Applications of $\mathcal{P}^2$-SRC

#### 4.5.1. Application I: cloud-enabled authentication

Authentication system on mobile devices identifies the individuals based on personal biometrics or passwords to grant access. Authentication based on face recognition is popularly used in mobile sensing applications. It recognizes the genuine user according to his facial appearance captured by the embedded cameras on the mobile devices. However, photos of human faces are very sensitive and people are reluctant to make such photos exposed to the public. Therefore, how to securely manage the collected face photos is crucial.

In the cloud-enabled face recognition system with $\mathcal{P}^2$-SRC, the end device, e.g., a smartphone, captures a photo of face of the subject to be authenticated. Then it compresses and encrypts the photo and uploads it to the cloud via anonymous communication protocol. The cloud computes an intermediate result by solving the most computationally intensive task and returns back this result to the smartphone. The final authentication decision will be made locally on the smartphone by solving lightweight residual computation in case the cloud gives a fake authentication decision to collude with an adversary to fool the smartphone system.

Our proposed $\mathcal{P}^2$-SRC enhances the security of authentication systems in two sides. On one hand, the compressed key encrypts face photos so that the adversaries cannot reconstruct the face photos without the compressed key. Even in case the cloud is compromised and the compressed key is disclosed, the face photos still cannot be reconstructed because the compressed data is not sufficient for an accurate reconstruction (as shown in the privacy analyses in Section 4.6.1, evaluations in Section 5.2.3 and user study II). On the other hand, $\mathcal{P}^2$-SRC also prevents adversaries from breaking into the mobile devices system when they collude with the cloud server. This is achieved by leveraging an anonymous communication protocol which protects the sources. For example, in cloud-enabled mobile authentication system, without the protection of sources, the compromised cloud server can deduce the possible sparse representation vector of the real user according to the records of historical authentication activities from the same "source" and return this "fake" sparse representation vector to the mobile devices when the adversaries attempt to get authenticated. Note that the labels and sources are equal in authentication system because the class labels are just the identities of the users.

We implement this application as a prototype of $\mathcal{P}^2$-SRC in Section 6 where we provide example of implementation details and evaluate its system cost.

#### 4.5.2. Application II: medical diagnosis classification system

Our proposed system is able to protect the privacy of users in privacy-sensitive applications (i.e., a medical diagnose system) where a trusted third party exists (i.e., a doctor). The label information is significantly sensitive when the interpretation of the classification results is related to the private information of the users (patients in this case). Medical diagnoses based on classification techniques are well studied in the literature. For instances, emotional facial expressions classification is often used in depression recognition [11,41,42]; activities and gait recognitions can be used to assist medical diagnosis of Alzheimer's disease [43,47]. In the medical diagnosis system, labels must be protected because the interpretation of the class labels are related to the health status of the patients.

Different from the authentication system, there should be a trusted third party (i.e., the doctor) in the medical diagnosis system who maintains the class labels information uploaded from the data contributors. With $\mathcal{P}^2$-SRC, the cloud only computes the sparse representation vector without the knowledge of class labels, therefore, in label privacy attack, the adversaries cannot infer the classification results from the sparse representation vector even if the cloud is compromised. In this scenario, the data contributors (some patients) are different from the application users (the doctors). The class labels can be used to infer the sensitive information such as the diseases of the patients or the expertise of the doctors.

### *4.6. Privacy analysis*

#### *4.6.1. Reconstruction attack*

*Reconstruction attack*, or termed as *content privacy attack* in this paper aims to estimate the original data by applying reconstruction techniques. In this section, we will prove that $\mathcal{P}^2$-SRC is resilient to the reconstruction attack meanwhile preserving the classification accuracy even if the compressed key is disclosed.

***Preliminary***. We assume that the cloud server has built a training set $\mathcal{A}$ consisting of encrypted data vectors from $P$ classes. The original data vector contains $N$ elements where $N$ is huge (we only consider the classification of high dimensional data in this paper). The compressed key $\mathcal{K}_1 \in \mathbb{R}^{m_1 \times N}$ ($m_1 \ll N$) maps the high dimensional data vectors into a significantly lower dimensional space. In the classification phase, a test data vector $x_1 \in \mathbb{R}^N$ is obtained and then encrypted to $y_1$ with the compressed key $\mathcal{K}$ (i.e., $y_1 = \mathcal{K} \cdot x_1$) on mobile device. $y_1$ is uploaded to the cloud server.

**Proposition 1.** *In a cloud-enabled mobile classification system where $P \ll N$, a compressed data vector $y_1$ can be resilient to reconstruction attack meanwhile produces high recognition accuracy when it is compressed using random projection matrix $\mathcal{K}$, if $\mathcal{O}(\log P) < m_1 \ll \mathcal{O}(s_r \log N)$.*

*Proof of Proposition 1.* We now give a theoretical proof for the performance of $\mathcal{P}^2$-SRC on the trade-off between privacy protection and classification accuracy under reconstruction attack according to the theory of Compressive Sensing [14].

*4.6.1.1. Privacy protection.* The performance of $\mathcal{P}^2$-SRC is determined by the number of projections $m_1$: larger $m_1$ brings higher classification accuracy, however, the compressed data vector $y_1$ is easier to be reconstructed, i.e., is vulnerable under reconstruction attack. In the reconstruction attack, we assume the cloud server is compromised. The adversary has the compressed data vector $y_1 \in \mathbb{R}^{m_1}$ and the compressed key $\mathcal{K}_1 \in \mathbb{R}^{m_1 \times N}$. The adversary wants to reconstruct the original data vector $x_1$ by solving the following $\ell_1$ optimization problem,

$$\hat{\theta}_r = \arg\min \|\theta_r\|_1 \quad \text{subject to } \|y_1 - \mathcal{K}_1 \Psi \theta_r\|_2 < \epsilon \tag{13}$$

Then the reconstructed data vector $\hat{x}_1 = \Psi \hat{\theta}_r$, where $\Psi$ is called the sparse basis or dictionary. It can be some standard orthonormal basis (e.g., Fourier transform basis or wavelet transform basis) or learned from some specific signals. We consider the worst situation where $\Psi$ is known (e.g., Fourier transform basis). According to the formal proof in [14], $x_1$ will be accurately reconstructed if the minimal number of projections $m_{att}$ satisfies,

$$m_{att} = \mathcal{O}(s_r \log N) \tag{14}$$

where $s_r$ is the sparsity of $x_1$ under the basis $\Psi$ therefore $s_r \geq 1$.

*4.6.1.2. Classification accuracy.* Correct classification can be achieved if we can obtain an accurate estimation of the sparse representation of $y_1$ under training set $\mathcal{A}$ via $\ell_1$ optimization presented in Eq. (10). Again, according to the theory of compressive sensing, an accurate sparse representation can be achieved when the minimal number of projections $m_c$ satisfies,

$$m_c = \mathcal{O}(s_1 \log P) \tag{15}$$

where $s_1$ is the sparsity of $y_1$ under training set $\mathcal{A}$. In most of the classification system, the sparsity $s_1$ should be equal to 1 because $y$ should belong to only one class. Therefore, it can be rewritten as $m_c = \mathcal{O}(\log P)$.

*Choosing of* m*1*. In the classification system with high dimensional data vectors, $N \gg P$. For example, in facial expression recognition system, the number of classes $P$ is below 10 (for example, the popular used Japanese Female Facial Expression dataset only contains 6 basic expressions and 1 neutral.) while the dimensionality of original data vectors will be over tens of thousands. According to the above statements, it is obvious that

$$s_r \log N \gg \log P$$

as $N \gg P$. So that $m_{att} \gg m_c$. It indicates that the number of projections required for the classification system is significantly less than that required for data reconstruction. As the gap between the two bonds are huge, the application publisher can easily choose a proper number of projections $m_1$ to provide an accurate classification service meanwhile protect the data vectors from reconstruction attack, i.e., let

$$m_c < m_1 \ll m_{att} \tag{16}$$

To verify our proof, we also conduct the dataset evaluations on two classification applications in Sections 5.2.3 and 5.3.3. The evaluation results show that $\mathcal{P}^2$-SRC produces high classification accuracy with relatively low number of projections and the attackers cannot obtain even a close approximation for the original data, e.g., faces or activity data which indicates both privacy and utility can be preserved.

#### *4.6.2. Collusion attack*

We consider two types of *collusion attacks* in our paper: the label privacy attacks and source privacy attacks. In collusion attack, the cloud server colludes with some of the mobile devices in the same classification application to infer the class labels and sources of the data vectors. In $\mathcal{P}^2$-SRC we achieve the source and label protections by leveraging anonymous communication channel, i.e., the Tor-like network, in an SRC-based framework. The protection of the sources can be achieved if at least one proxy is not compromised which indicates higher level of privacy protection can be achieved by adding new proxy [8]. However, to avoid the overwhelming system cost, the number of proxies are decided by the level of privacy protection required in $\mathcal{P}^2$-SRC. To determine the number of proxies needed, we adopt a probability-based Tor-like network scheme. Considering a system with $p$% of malicious users who would potentially collude with the cloud server. The application user randomly selects $n_m$ mobile devices in the system as a cascade of proxies. The probability that at least one proxy in the cascade is trustworthy (i.e., the sources can be protected) therefore is $1 - p^{n_m}$, the probability is also equivalent to the probability that the source is protected. For example, if 50% of the users in the system is malicious, and we choose a cascade of five users, then $1 - 0.5^5 = 0.97$, hence the probability that the source is protected is 0.97. Moreover, as the uploading trace is determined locally, the source mobile devices are convenient to change their uploading traces to enhance security.

For the protection of labels, as computing sparse representation does not need the information of class labels, data contributors can only upload the encrypted data vectors and hide their class labels from the cloud server. Therefore the label privacy leakage can be avoided.

## 5. Dataset evaluation

### 5.1. Goals, metrics, and methodology

The aim of this section is to evaluate the performance of $\mathcal{P}^2$-SRC on classification accuracy and privacy protection. For the classification accuracy, we compare $\mathcal{P}^2$-SRC with original SRC approach, Pickle [25] and Nearest Neighborhood (NN). We compare with NN (1-NN) but not K-NN because 1-NN has been proved at least as good as K-NN for classification problem [48,54]. For the privacy-preserving, we compare $\mathcal{P}^2$-SRC with the traditional random noise perturbation approach. We evaluate the performance of $\mathcal{P}^2$-SRC in two different classification applications: face recognition and activity recognition. The evaluations are made on two publicly available datasets: the Extended Yale Dataset B [16] and UCI Human Activity Recognition Using Smartphones Dataset [3]. At last we also conduct two user studies to provide more intuitive judgement on the performance of the privacy protection and users' attitudes towards the sensitivity of different types of privacy attacks.

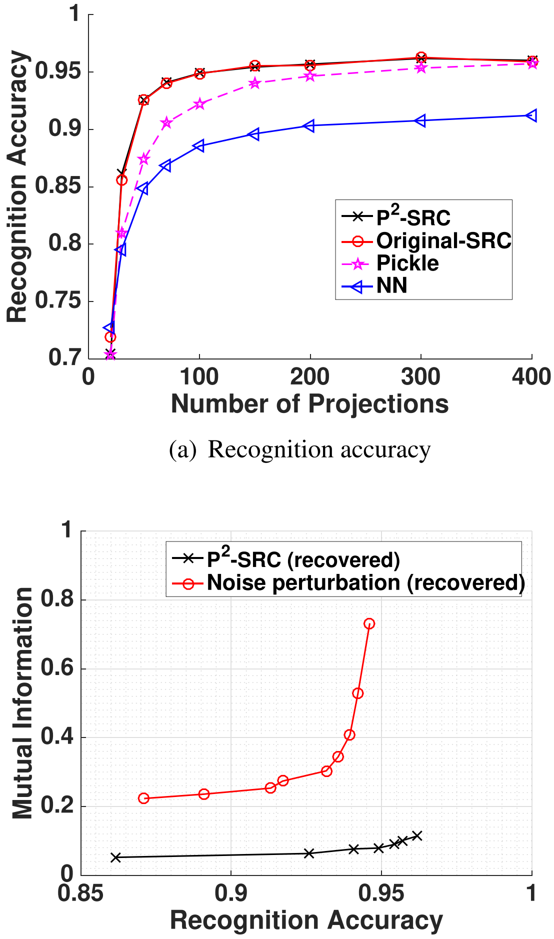
For a fair comparison, we apply the same preprocessing, i.e., normalizing all the data samples, to the datasets for all comparing algorithms. We also use variables control strategy. For example, when comparing the recognition accuracy of $\mathcal{P}^2$-SRC with other classification algorithms, we first normalize all the data vectors in the datasets then use random projections as the input features and the only difference is the classifiers they adopt. It is a reasonable setting because random projections are generally applicable input features for different classifiers as the evaluation by Wright et al. [48], including SRC, Supportive Vector Machine (SVM), NN and Nearest Subspace (NS). Meanwhile we do not consider the performance of privacy preserving when evaluate the recognition accuracy. However when comparing trade-off between the privacy preserving and the recognition accuracy, SRC is adopted as the classifier for both $\mathcal{P}^2$-SRC and random noise perturbation to exclude the influence of different classifiers used.

In this paper, we use the percentage of correct recognition as the performance metric of the recognition accuracy, which is the number of right recognition over the total number of tests. Then we evaluate the performance of privacy protection under reconstruction attack. We use *MI-RA* (mutual information against recognition accuracy) and *Coherence-RA* (coherence against recognition accuracy) curves to estimate the privacy protection under different recognition accuracies achieved. Lower MI or Coherence between the reconstructed and original data vectors represents better privacy protection achieved.

### 5.2. Face recognition

#### 5.2.1. Dataset description

Extended Yale Dataset B consists of 38 subjects under 9 poses and 64 illumination conditions. For each trail of the evaluation, we randomly pick 30 face images from each subject to form the dictionary $\mathcal{D}$ and the rest of the images are used as the tests. The data vector is derived from concatenating the pixels of the image by rows. As the resolution of the face images is $192 \times 168$ in the dataset, each data vector contains 32, 256 elements and it is significantly larger than the number of classes (i.e., 38). Therefore the size of dictionary $\mathcal{D}$ is $32,256 \times 1,140$. To simulate $\mathcal{P}^2$-SRC using this dataset, we randomly shuffle the sequences of the columns in $\mathcal{D}$. During the test phase, each class only knows locations of its own in the training set. While in the simulation of traditional SRC method, the face images from the same class are grouped together and the information of all the class labels is public.



(a) Recognition accuracy

(b) Mutual information

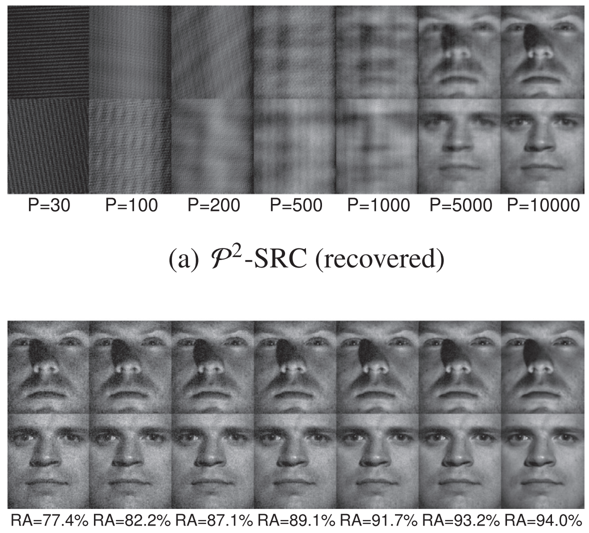
**Fig. 2.** Results of face recognition.

#### 5.2.2. Recognition accuracy

To show that $\mathcal{P}^2$-SRC provides high face recognition accuracy, we compare $\mathcal{P}^2$-SRC with original SRC, Pickle and NN methods. Compressed key is randomly generated from Gaussian distribution and is used to compress and encrypt the data vectors. We gradually change the number of projections from 10 to 400 and compute the overall recognition accuracy over the 38 subjects. The resultant face recognition accuracy is represented by the average over 30 independent trails where different training sets are used. As the results shown in (a), we can find $\mathcal{P}^2$-SRC achieves almost the same recognition accuracy compared with original SRC method and it also outperforms Pickle. Although its accuracy is only 4% higher than Pickle when the number of projections is 100, it protects more types of privacies (source and label) than Pickle. Another observation is the growth of the recognition accuracy diminishes when the number of projections is over 100. Considering the fact that larger number of projections leads to more energy consumption on the local mobile devices due to computation and data transmission, 100 is chosen as the number of projections for $\mathcal{P}^2$-SRC in face recognition application implementation.

#### 5.2.3. Privacy evaluation

Considering the situation where the cloud server is compromised, adversaries obtain the compressed key and undertake reconstruction attack. The metric we use to evaluate the information disclosure under the reconstruction attack is the *Mutual Information* (MI). MI is popularly used to estimate the accuracy of image



(a) $\mathcal{P}^2$-SRC (recovered)

(b) Noise perturbation (recovered)

**Fig. 3.** Samples of Reconstructed faces from different encryption methods.

reconstruction methods and is defined as

$$MI(f_1, f_2) = En(f_1) + En(f_2) - JointEn(f_1, f_2) \tag{17}$$

where $f_1$ and $f_2$ are two images, $En(\cdot)$ is the entropy and $jointEn(\cdot)$ is the joint Entropy. Then we scale the value of MI to $0-1$. To present $\mathcal{P}^2$-SRC is resilient to the reconstruction attack even if the compressed key is disclosed, we compare it with the traditional random noise perturbation strategy. The noise added are randomly drawn from a zero-mean-one-norm Gaussian distribution. The results are obtained from the average over 100 face images and represented by *MI-RA* trade-off curve. The *MI-RA* presents the mutual information between the reconstructed image and original image against the recognition accuracy achieved. We use *MI-RA* curve because the higher recognition accuracy achieved, the more information will be disclosed. $\ell_1$ optimization (as Eq. (13)) is used to reconstruct the face images encrypted by $\mathcal{P}^2$-SRC; Principal Component Analysis (PCA) is used to reconstruct the face images protected by the noise perturbation approach.

As the results shown in Fig. 2(b), we can find $\mathcal{P}^2$-SRC is significantly more resilient to the reconstruction attack compared with the traditional noise perturbation approach when the same face recognition accuracy is achieved. For instance, when the face recognition accuracy is around 90%, MI between the original and reconstructed face image is only 0.04 for $\mathcal{P}^2$-SRC but 0.24 for noise perturbation.

To provide more intuitive results to the readers, we present the samples of reconstructed face images of two individuals. In Fig. 3(a) and Fig. 3(b), we present two rows of samples and each row consists of reconstructed face images from one individual. As the samples shown in Fig. 3(a), we can see that the faces are not recognizable by human until the number of projections $p$ reaches $1000-5000$ which is significantly more than the number of projections used in our $\mathcal{P}^2$-SRC face recognition system ($p = 100$). The corresponding face image sample when $p = 100$ in Fig. 3(a) demonstrates that our system is resilient to reconstruction attack when $p = 100$ and we actually cannot identify any shape of faces in the images where the corresponding recognition accuracy has reached 95%. However, as the face image samples shown in Fig. 3(b), face images protected by noise perturbation can be easily reconstructed via PCA approach and we can still clearly recognize the reconstructed faces even though the recognition accuracy (RA) drops to only 77.4%. Therefore we can claim that our $\mathcal{P}^2$-SRC face recognition system is resilient to the reconstruction attack even if the compressed key is disclosed.

### 5.3. Activity recognition

#### 5.3.1. Dataset description

The UCI Human Activity Recognition Using Smartphones Dataset was collected from a group of 30 volunteers within the age of $19-48$. Each person performs 6 activities wearing a smartphone on their waist. therefore the number of classes in this dataset is 6. Using the embedded accelerometer and gyroscope, 561-dimensional feature vector is extracted by calculating variables from the time and frequency domain (details about the types of the features can be found in [3]). Again, the dimensionality of the feature vectors ($N = 561$) is significantly larger than the number of classes ($P = 6$). The collected dataset is divided into two parts. 70% of the participants contribute data for the training part while the rest of the participants contribute data for the test part. In our evaluation for activity recognition, we randomly select 100 feature vectors of each activity from the training part to form the dictionary $\mathcal{D} \in \mathbb{R}^{561\times600}$. Therefore the size of the dictionary $\mathcal{D}$ is $561 \times 600$. Then we randomly select 100 feature vectors from the test part to form the test set. Again, to simulate $\mathcal{P}^2$-SRC, the sequence of the columns in $\mathcal{D}$ is shuffled.

#### 5.3.2. Recognition accuracy

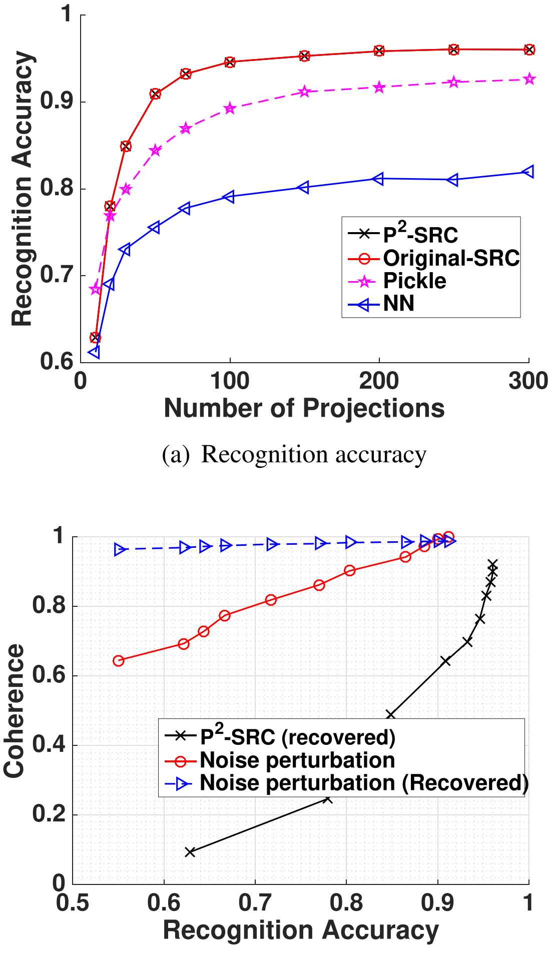
To present $\mathcal{P}^2$-SRC activity recognition system achieves good recognition accuracy, we again compare it with original SRC, Pickle and NN methods. We use random Gaussian matrix as the compressed key to compress and encrypt the feature vectors. During the simulation, we gradually change the number of projections from 10 to 300 and represent the recognition accuracy over averaging the results from 30 independent trails where different training sets and test sets are randomly selected. As the recognition accuracy shown in Fig. 4(a), $\mathcal{P}^2$-SRC activity recognition system achieves exactly the same recognition accuracy to the traditional SRC approach because in this application scenario, the trusted third party has the full information of class labels. For example, the recognition accuracy of $\mathcal{P}^2$-SRC is around 95% which is the same to traditional SRC while Pickle and NN methods is only around 88% and 77% when $p = 70$.

#### 5.3.3. Privacy evaluation

Then we compare the performance of privacy protection of $\mathcal{P}^2$-SRC and noise perturbation approach against the reconstruction attack when the compressed key is disclosed. The metric we use to evaluate the information disclosure for the reconstruction attack on the feature vectors of the activities is the *Coherence* between the reconstructed feature vector and its corresponding original feature vector. The Coherence is scaled to $0-1$ and larger coherence indicates the reconstructed feature vector is more similar to the original feature vector.

The results of reconstruction attack are evaluated over 600 feature vectors (100 feature vectors for each class) and shown in Fig. 4(b). Besides the results of $\mathcal{P}^2$-SRC and noise perturbation after signal reconstruction, we also present the results of the noise perturbation without reconstruction as the baseline. From Fig. 4(b) we can find $\mathcal{P}^2$-SRC achieves the best performance on privacy protection against reconstruction attack. The coherence produced by $\mathcal{P}^2$-SRC is significantly smaller than that of the noise perturbation approach without reconstruction.

To provide more intuitive results, we present some samples of feature vectors and their reconstructed vectors from different privacy protection schemes (only the first 100 data points are drawn in the figure for the page size limit). In Fig. 5, the first row of



(a) Recognition accuracy

(b) Coherence

**Fig. 4.** Results of activity recognition.

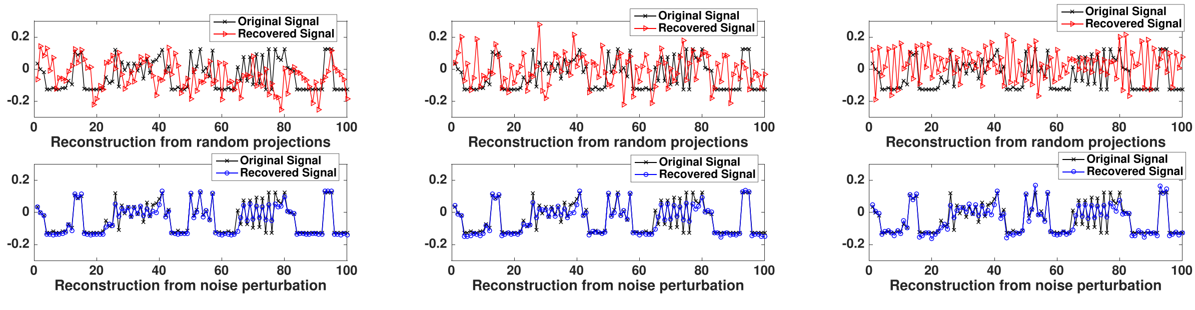
figures presents the reconstructed feature vector values from $\mathcal{P}^2$-SRC while the second row is the reconstructed feature vector values from noise perturbation approach. Different columns from left to right are marked with their corresponding recognition accuracy (RA). The original feature vector values are also included in the sub-figures as the reference. From the appearance of Fig. 5, it is obvious that the reconstructed feature vectors in the first row are significantly different from the original feature vectors. It indicates $\mathcal{P}^2$-SRC protects the privacy of the feature vectors against reconstruction attack well. However, as the results shown in the second row, the feature vectors with noise perturbation are accurately reconstructed, the reconstructed feature vectors match the original feature vectors very well. Therefore, noise perturbation approach discloses the information of the feature vectors under reconstruction attack.

### *5.4. User studies*

In this section we will discuss the results from two user studies to 1) show the sensitivity of users to content, source and label privacy attacks and 2) evaluate the performance of $\mathcal{P}^2$-SRC under reconstruction attack. We recruit 100 undergraduates to participate in two questionnaire survey based user studies.

*5.4.1. User study I.* During the first user study, the users are presented four sequences of reconstructed face images similar to those in Fig. 3. The number of projections used in $\mathcal{P}^2$-SRC to derive the compressed face images vary from 30 to 10, 000. The recruited users mark each of the face images **1, 2** or **3** where **1** means the users cannot see any **face** from the image, **2** means users can find some **face** from the image but they consider the **face** is not clear enough to be recognized and **3** means users may identify the owner of the face from this image. The average score over 100 users are computed and shown in Fig. 6(a). It is obvious that users cannot recognize the identities of the faces when the number of projections are below 1, 000. According to Fig. 2(a), 100 projections are sufficient for $\mathcal{P}^2$-SRC to achieve accurate face recognition however, all of the users agree that **there is no face in the image** under the 100 projections setting.

*5.4.2. User study II.* In this user study, we investigate the sensitivity (importance) of different types of privacy attacks (content, label and source) to users. We provide a questionnaire survey to the 100 recruited users. The users determine their sensitivity to four different scenarios including the case history of the physical diseases (scenario #1), the recording of conversation with psychologist (scenario #2), audio recording of their ambient environment (scenario #3) and sensor readings of the smartphones generated from their daily activities (scenario #4). The sensitivity is represented by a score from 0 to 10 where 10 stands for the highest sensitivity. We compute the average score of sensitivity of the three types of privacies and present the results in Fig. 6(b). The results demonstrate the subjective feelings of users towards different application scenarios and types of privacy attacks. It is clear that the sensitivity of the users varies with different scenarios meanwhile the users tend to concern more about the source and label attacks than content though content privacy has been studied intensively in existing literature of privacy-preserving CMAs.

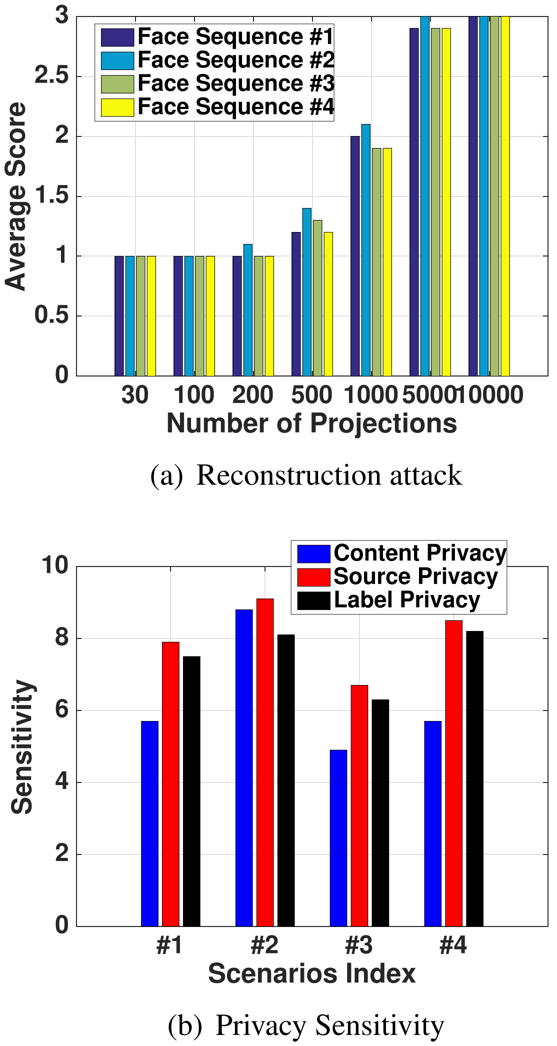


(a) RA≈ 90%

(b) RA≈ 86%

(c) RA≈ 77%

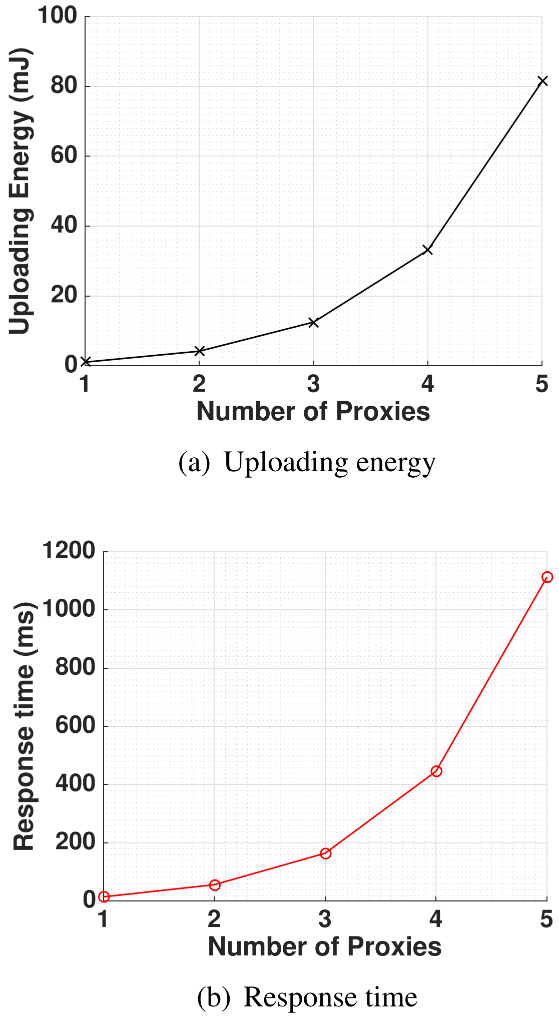
**Fig. 5.** Samples of Reconstructed activity signals from different encryption methods.



(a) Reconstruction attack

(b) Privacy Sensitivity

**Fig. 6.** Results of user studies.



(a) Uploading energy

(b) Response time

**Fig. 7.** Energy consumption of $\mathcal{P}^2$-SRC.

## 6. System evaluation

To evaluate the system cost of $\mathcal{P}^2$-SRC on mobile devices, we implemented a prototype of cloud-enabled face recognition system basing on $\mathcal{P}^2$-SRC on smartphones and conducted face recognition experiments.[1] 10 subjects were recruited as the data contributors (they were also application users) and each of the subjects contributed 20 face images with resolution of 192 × 168 during training set collection phase. Therefore the size of the dictionary $\mathcal{D}$ was 32, 256 × 200. The face images were compressed and encrypted to data vector of 100 projections according the evaluation on the face dataset in Section 5.2.2.

The system was implemented on 10 off-the-shelf Android smartphones running Android 4.4.4 (4 sets of Samsung Galaxy Note 4, 4 sets of Samsung Galaxy S6 and 2 sets of Samsung Galaxy S5) and one Macbook pro laptop running Mac OS X EI Capitan (specifications: 2.5 GHz Intel i7 CPU, 16GB RAM and 512GB SSD) as the cloud server. Wi-Fi was used for data transmission among mobile devices and cloud server. It is worth noting that our system is just a demonstration for the feasibility of the prototype and it is implemented within a local area network. However, $\mathcal{P}^2$-SRC is orthogonal to the network services. For examples, the scale of the system can be extended by adopting mobile data services, e.g., LTE; the anonymization of identities can be achieved by purchasing the current available anonymous communication services, e.g., The Onion Router (TOR). The cost of the operation could be shared by the app publisher and users and the data contributors could earn credit by contributing their data. However, the detailed investigation is beyond the scope of this paper.

We evaluated the energy consumption of the data encryption and wireless communication on mobile devices. The energy consumption was calculated using Android APIs [1]. In our experiment, similar face recognition accuracy was achieved as the evaluation in Section 5.2.2, but the figure of the recognition accuracy is not presented due to the page limit. As the recognition accuracy has been well evaluated in the previous section, we emphasize on the evaluation of system cost in this section.

We changed the number of proxies (intermediate mobile devices) from 1 to 5 and the number of projections in each data vector was 100 to evaluate the energy consumption of the mobile devices when uploading data vector from data contributor to the cloud server. As the results shown in Fig. 6(a), the energy consumption increased with the growth of the number of proxies. For example, when 5 proxies were used in the Tor-like network, total energy consumption of the 6 mobile devices (5 proxies and 1 data contributor) was around 83 mWs (mJ). The average energy consumption of each mobile device was around 14 mWs (mJ) for uploading each data vector. The battery capacity of common mod-

[1] Ethical approval is granted by Massachusetts Institute of Technology (Reference Number 1502006877).

ern smartphones is around 30 kWs (KJ), therefore, the energy cost of each uploading only accounts to around 0.000041% of the total energy supply. We assume the smartphone has a targeted lifespan of one day, which results in an energy budget of 1.25 KWs (KJ) per hour. To put this into perspective, with only 1% of the budget per hour, i.e., 12.5 Ws (J), $\mathcal{P}^2$-SRC is able to perform uploading operations approximately 893 times per hour, i.e., upload around 15 data vectors every minute.

The application user uploaded both the compressed test vector and returning trace to the cloud server via the Tor-like network. When the cloud server received the test vector, it computed the sparse representation of the test vector. Then the 200-dimensional sparse representation vector was returned to the application user. We evaluated the average response time of the cloud-enabled classification via different number of proxies (from 1 to 5). As the results shown in Fig. 6(b), the response time increased with the growth of the number of proxies traversed. When 5 proxies are used, the total response time to undertake one classification request with $\mathcal{P}^2$-SRC was about 1.1 s. Though the system delay seems non-negligible, considering the fact $\mathcal{P}^2$-SRC achieves significantly higher level of privacy than previous approaches, it is worth reasonable sacrifice on time delay for highly-sensitive applications. Meanwhile, the system delay can be tuned by changing the number of proxies to form the Tor-like network. For example, when the mobile users are more trustworthy, we can largely reduce the number of proxies involved to reduce the system delay.

## 7. Conclusion

In this paper, we propose a new privacy-preserving framework, $\mathcal{P}^2$-SRC, for classification in cloud-enabled mobile applications. $\mathcal{P}^2$-SRC is outstanding of the existing solutions by addressing different types of privacy attacks including content privacy attacks, class label privacy attacks and source privacy attacks. As the evaluations with different classification applications, user studies on a large group of participants and real world system implementation, $\mathcal{P}^2$-SRC produces the best trade-off between the privacy protection and recognition accuracy under reconstruction attack and the system cost introduced is acceptable.

## Acknowledgement

This work is partially supported by National Natural Science Foundation of China under Grant 61702133 and 61702132, Natural Science Foundation of Heilongjiang province under grant QC2017069, QC2017071 and QC2016083, the Fundamental Research Funds for the Central Universities under Grant HEUCFJ160601 and HEUCFJ170602, the China Postdoctoral Science Foundation under Grant 166875 and Heilongjiang Postdoctoral Fund under Grant LBH-Z16042. The authors are grateful to the reviewers and editors for their intensive reviews and insightful comments to improve the quality of this manuscript.

## References

[1] Power profiles for android, 2017.
[2] P. Aditya, R. Sen, P. Druschel, S.J. Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, T.T. Wu, I-pic: A platform for privacy-compliant image capture, MobiSys'16, 16, 2016.
[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, 2013.
[4] X. Bao, R. Roy Choudhury, Movi: mobile phone based video highlights via collaborative sensing, in: Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, 2010, pp. 357–370.
[5] M.N. Burns, M. Begale, J. Duffecy, D. Gergle, C.J. Karr, E. Giangrande, D.C. Mohr, Harnessing context sensing to develop a mobile intervention for depression, J. Med. Internet Res. 13 (3) (2011).
[6] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, Privacy-preserving multi-keyword ranked search over encrypted cloud data, Parallel Distrib. Syst. IEEE Trans. 25 (1) (2014) 222–233.
[7] S. Chakraborty, C. Shen, K.R. Raghavan, Y. Shoukry, M. Millar, M. Srivastava, ipshield: a framework for enforcing context-aware privacy, in: 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), USENIX Association, 2014, pp. 143–156.
[8] D.L. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (1981) 84–90.
[9] S. Chen, A. Pande, P. Mohapatra, Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones, in: Proceedings of the 12th annual international conference on Mobile systems, applications, and services, ACM, 2014, pp. 109–122.
[10] S.S.M. Chow, J.H. Lee, L. Subramanian, Two-party computation model for privacy-preserving queries over distributed databases, Network and Distributed System Security Symposium, 2009.
[11] J.F. Cohn, T.S. Kruez, I. Matthews, Y. Yang, M.H. Nguyen, M.T. Padilla, F. Zhou, F.D. La Torre, Detecting depression from facial actions and vocal prosody, in: Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on, IEEE, 2009, pp. 1–7.
[12] Y.S. Dan Yin, Location and Relation Based Clustering on Privacy-preserving Social Networks, Tsinghua Sci. Technol.(0) 85. doi: 10.26599/TST.2018.9010017.
[13] R. Dingledine, N. Mathewson, P. Syverson, Tor: the second-generation onion router, J. Franklin Inst. 239 (2) (2004) 135–139.
[14] D. Donoho, Compressed sensing, TIT (2006) 1289–1306.
[15] R.K. Ganti, N. Pham, Y.-E. Tsai, T.F. Abdelzaher, Poolview: stream privacy for grassroots participatory sensing, in: Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM, 2008, pp. 281–294.
[16] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, PAMI 23 (6) (2001) 643–660.
[17] R. Herbster, S. DellaTorre, P. Druschel, B. Bhattacharjee, Privacy capsules: Preventing information leaks by mobile apps, in: Proc. of MobiSys, 2016.
[18] P. Hornyack, S. Han, J. Jung, S. Schechter, D. Wetherall, These aren't the droids you're looking for: retrofitting android to protect data from imperious applications, in: Proceedings of the 18th ACM conference on Computer and communications security, ACM, 2011, pp. 639–652.
[19] J. Howe, The rise of crowdsourcing, Wired Mag. 14 (6) (2006) 1–4.
[20] K. Huang, S. Aviyente, Sparse representation for signal classification, in: Advances in neural information processing systems, 2006, pp. 609–616.
[21] P. Huang, T. Xu, X. Jin, Y. Zhou, Defdroid: Towards a more defensive mobile os against disruptive app behavior.
[22] H. Khan, U. Hengartner, D. Vogel, Targeted mimicry attacks on touch input based implicit authentication schemes, in: MobiSys'16, ACM, 2016, pp. 387–398.
[23] A. Krause, C. Guestrin, Optimizing sensing: from water to the web, Computer 42 (2009) 38–45.
[24] T.-S. Lim, W.-Y. Loh, Y.-S. Shih, A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, Mach. Learn. 40 (3) (2000) 203–228.
[25] B. Liu, Y. Jiang, F. Sha, R. Govindan, Cloud-enabled privacy-preserving collaborative learning for mobile sensing, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, ACM, 2012, pp. 57–70.
[26] H. Lu, W. Pan, N.D. Lane, T. Choudhury, A.T. Campbell, Soundsense: scalable sound sensing for people-centric applications on mobile phones, in: Proceedings of the 7th international conference on Mobile systems, applications, and services, ACM, 2009, pp. 165–178.
[27] C. Luo, M.C. Chan, Socialweaver: collaborative inference of human conversation networks using smartphones, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ACM, 2013, p. 20.
[28] H. Ma, D. Zhao, P. Yuan, Opportunities in mobile crowd sensing, Commun. Mag. IEEE 52 (8) (2014) 29–35.
[29] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, K. Ren, Cloud-enabled privacy-preserving truth discovery in crowd sensing systems, SenSys'15, ACM, 2015.
[30] E. Miluzzo, C.T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, A.T. Campbell, Darwin phones: the evolution of sensing and inference on mobile phones, in: Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, 2010, pp. 5–20.
[31] S. Mirzamohammadi, A.A. Sani, Viola: Trustworthy sensor notifications for enhanced privacy on mobile systems, MobiSys'16, 2016.
[32] A. Panchenko, F. Lanze, T. Engel, Improving performance and anonymity in the tor network, in: IEEE International Performance Computing and Communications Conference (IPCCC),, 2012, pp. 1–10.
[33] A. Parate, M.-C. Chiu, D. Ganesan, B.M. Marlin, Leveraging graphical models to improve accuracy and reduce privacy risks of mobile sensing, in: Proceeding of the 11th annual international conference on Mobile systems, applications, and services, ACM, 2013, pp. 83–96.
[34] Y. Rachlin, D. Baron, The secrecy of compressed sensing measurements, in: Communication, Control, and Computing, 2008 46th Annual Allerton Conference on, IEEE, 2008, pp. 813–817.
[35] N. Raval, A. Srivastava, A. Razeen, K. Lebeck, A. Machanavajjhala, L.P. Cox, What you mark is what apps see, MobiSys'16, 2016.
[36] K. Schramm, G. Leander, P. Felke, C. Paar, A collision-attack on aes, in: Cryptographic Hardware and Embedded Systems-CHES 2004, Springer, 2004, pp. 163–175.

[37] Y. Shen, W. Hu, J. Liu, M. Yang, B. Wei, C.T. Chou, Efficient background subtraction for real-time tracking in embedded camera networks, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, ACM, 2012, pp. 295–308.

[38] Y. Shen, W. Hu, M. Yang, J. Liu, B. Wei, S. Lucey, C.T. Chou, Real-time and robust compressive background subtraction for embedded camera networks, IEEE Trans. Mob. Comput. 15 (2) (2016) 406–418.

[39] Y. Shen, W. Hu, M. Yang, B. Wei, S. Lucey, C.T. Chou, Face recognition on smartphones via optimised sparse representation classification, in: Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on, IEEE, 2014, pp. 237–248.

[40] Y. Shen, M. Yang, B. Wei, C.T. Chou, W. Hu, Learn to recognise: exploring priors of sparse face recognition on smartphones, IEEE Trans. Mob. Comput. 16 (6) (2017) 1705–1717.

[41] S.A. Surguladze, A.W. Young, C. Senior, G. Brébion, M.J. Travis, M.L. Phillips, Recognition accuracy and response bias to happy and sad facial expressions in patients with major depression., Neuropsychology 18 (2) (2004) 212.

[42] T. Suslow, K. Junghanns, V. Arolt, Detection of facial expressions of emotions in depression, Percept. Mot. Skills. 92 (3) (2001) 857–868.

[43] J. Verghese, R.B. Lipton, C.B. Hall, G. Kuslansky, M.J. Katz, H. Buschke, Abnormality of gait as a predictor of non-alzheimer's dementia, N. Engl. J. Med. 347 (22) (2002) 1761–1768.

[44] B. Wei, W. Hu, M. Yang, C.T. Chou, Radio-based device-free activity recognition with radio frequency interference, in: Proceedings of the 14th International Conference on Information Processing in Sensor Networks, ACM, 2015, pp. 154–165.

[45] B. Wei, M. Yang, Y. Shen, R. Rana, C.T. Chou, W. Hu, Real-time classification via sparse representation in acoustic sensor networks, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ACM, 2013, p. 21.

[46] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inf. Sci. 258 (2014) 371–386.

[47] R.S. Wilson, C.F.M. De Leon, L.L. Barnes, J.A. Schneider, J.L. Bienias, D.A. Evans, D.A. Bennett, Participation in cognitively stimulating activities and risk of incident alzheimer disease, Jama 287 (6) (2002) 742–748.

[48] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, Pattern Anal. Mach. Intell. IEEE Trans. 31 (2) (2009) 210–227.

[49] W. Xu, G. Revadigar, C. Luo, N. Bergmann, W. Hu, Walkie-talkie: Motion-assisted automatic key generation for secure on-body device communication, in: IPSN'16, IEEE, 2016, pp. 1–12.

[50] W. Xu, Y. Shen, N. Bergmann, W. Hu, Sensor-assisted face recognition system on smart glass via multi-view sparse representation classification, in: IPSN'16, IEEE, 2016, pp. 1–12.

[51] W. Xu, Y. Shen, N. Bergmann, W. Hu, Sensor-assisted multi-view face recognition system on smart glass, IEEE Trans. Mob. Comput. 17 (1) (2018) 197–210.

[52] W. Xu, Y. Shen, Y. Zhang, N. Bergmann, W. Hu, Gait-watch: A context-aware authentication system for smart watch based on gait recognition, in: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, ACM, 2017, pp. 59–70.

[53] Z. Yan, W. Ding, V. Niemi, A.V. Vasilakos, Two schemes of privacy-preserving trust evaluation, Future Gener. Comput. Syst. 62 (2016) 175–189.

[54] D. Yu, X. Yu, Q. Hu, J. Liu, A. Wu, Dynamic time warping constraint learning for large margin nearest neighbor classification, Inf. Sci. 181 (13) (2011) 2787–2796.

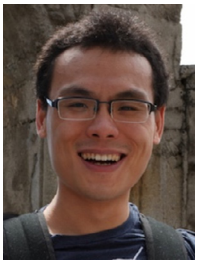
[55] J. Zhou, Z. Cao, X. Dong, A.V. Vasilakos, Security and privacy for cloud-based iot: challenges, IEEE Commun. Mag. 55 (1) (2017) 26–33.

[56] S. Zhu, L. Lu, K. Singh, Case: Comprehensive application security enforcement on cots mobile devices.

**Yiran Shen** received the PhD degree in computer science and engineering from the University of New South Wales. He is an associate professor in the College of Computer Science and Technology, Harbin Engineering University (HEU). He was SMART Scholar at Singapore-MIT Alliance for Research and Technology before he joined HEU. He publishes regularly at top-tier conferences and journals. His current research interests are wearable/ mobile computing, wireless sensor networks and applications of compressive sensing. He is a member of the IEEE.

**Chengwen Luo** received the PhD degree from the School of Computing, National University of Singapore, Singapore. He is currently an assistant professor in the College of Computer Science and Software Engineering, Shenzhen University (SZU), China. Before joining SZU, he was a postdoctoral researcher in CSE, The University of New South Wales, Australia. He is the author and co-author of several research papers in top venues of mobile computing and WSN such as ACM SenSys, ACM/IEEE IPSN, etc. His research interests include mobile and pervasive computing, indoor localization, wireless sensor networks, and security aspects of Internet of Things.

**Dan Yin** received her PhD degree in the department of computer science and technology at Harbin Institute of Technology. Dr. Yin is currently an Assistant Professor at Harbin Engineering University. Her research areas focus on big data, graph mining and privacy.



**Hongkai Wen** is an Assistant Professor in Department of Computer Science, University of Warwick. Before that he obtained his D.Phil at the University of Oxford, and became a post-doctoral researcher in a joint project between Oxford Computer Science and Robotics Institute. Broadly speaking, his research belongs to the area of Cyber-Physical Systems, which use networked smart devices to sense and interactive with the physical world.

**Daniela Rus** is a Professor of Electrical Engineering and Computer Science and Director of the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT. Prior to her appointment as Director, she served as Associate Director of CSAIL from 2008 to 2011, and as the Co-Director of CSAIL's Center for Robotics from 2005 to 2012. She also leads CSAIL's Distributed Robotics Laboratory. Rus is the first woman to serve as director of CSAIL, and its predecessors the AI Lab and the Lab for Computer Science.

**Wen Hu** is a senior lecturer at the School of Computer Science and Engineering, the University of New South Wales (UNSW). Much of his research career has focused on the novel applications, low power communications, security and compressive sensing in sensor network systems, and Internet of Things (IoT). He is a senior member of the IEEE.