数据隐私方法伦理和实践
Methodology, Ethics and Practice of Data Privacy

隐私保护的应用
Applications

张兰
中国科学技术大学 计算机学院
2020春季

1. PPML by HE

Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing

Jiawei Yuan, Shucheng Yu
13’TPDS

Contribution

- An efficient and scalable solution that supports collaborative BPN network learning with privacy preservation in the multi-party setting and allows arbitrarily partitioned datasets.

Main Idea

- Each participant first encrypts her/his private data with the system public key and then uploads the ciphertexts to the cloud
- Cloud servers then execute most of the operations pertaining to the learning process over the ciphertexts and then return the encrypted results to the participants
- The participants jointly decrypt the results with which they update their respective weights for the BPN network.
- Cloud servers learn no privacy data. Though off-loading the computation tasks to the cloud, this scheme is high scalable.

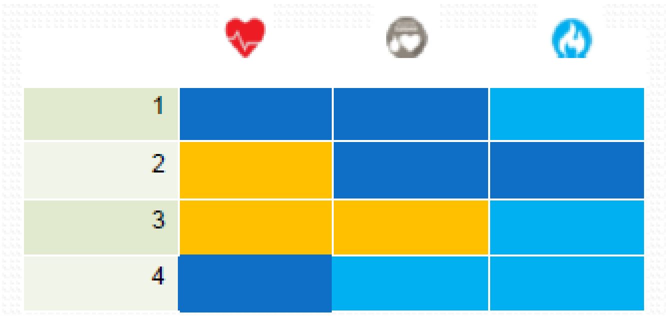
System Model

- Trusted authority (TA)
  - Generate and issue encryption/decryption keys
- Data owners
  - P_s owns a private data set and wants to perform collaborative learning with other data owners
- Cloud server
  - Execute most of the operations pertaining to the learning

Data Partition

» Data is arbitrarily partitioned among multi-parties K

- Assume the aggregated training set D with N records in total, each record with m attributes
- Each part P_s holds part of data set without specific order
- For example (3 parties)



BGN

Simultaneously supports one multiplication and unlimited number of addition operations.

$$C(m_1\hat{m}_1 + m_2\hat{m}_2 + \cdots + m_i\hat{m}_i)$$

- Private key: $SK = q$
- Public key: $PK = (n, G, G_1, e, g, h)$
  - e: $G \times G \rightarrow G_1$; generator: $g, u\ and\ set\ h = u^r, q \cdot r = n$
  - Define two cyclic G, G_1 whose order is n and q
- Encrypt – $C = g^m h^r \in G$
- Decrypt – $C^q = (g^m h^r)^q = (g^q)^m\ mod\ n$
  - Compute log of C^q base g^q (solving discrete logarithm of the ciphertext using Pollard's lambda method)
  - BGN scheme just works with small numbers

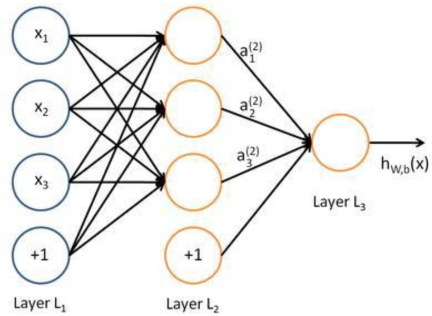
Scheme

1. Each party encrypt her/his input data set by BGN public key and upload the encrypted data to the cloud.
2. The cloud servers performs most of the operations, i.e., secure scalar product and addition.
3. As the BGN algorithm just supports one step multiplication over ciphertext, the intermediate results shall be first securely decrypted and then encrypted to support consecutive multiplication operations.
4. The decrypted results known to each party cannot be the actual intermediate values. We design a secret sharing algorithm that allows the parties to decrypt only the random shares of the intermediate values.
5. Sigmoid function approximation.

After the entire process of the privacy preserving learning, all the parties jointly establish a neural network representing the whole data set without disclosing any private data to each other.

How to apply HE (BGN)

- Parties
  - Encrypt her/his data with same public key
- Cloud server
  - Encrypt weight with public key
  - Compute scalar product on ciphertext, $C(L_s)$
  - Secure share the $C(L_s)$
- Parties
  - Compute sigmoid function
- Cloud server
  - Compute error on ciphertext (true label and output)
  - With the help of parties compare error and threshold
  - Update weight on ciphertext (true label and output and intermediated result) and Secure share the result



Cloud Secure Computing

- Encryption: Given a message M, encrypt it as: $C = g^M h^r \in G, r \xleftarrow{R} Z_n$
- Secure Scalar Product: Given the ciphertexts of vector $(M_{11}, M_{12}, \ldots, M_{1v})$ and $(M_{21}, M_{22}, \ldots, M_{2v})$ the cloud computes their scalar products as :

$$C(prod) = h_1^1 * \prod_{i=1}^{v} e(C_{1i}, C_{2i})$$

where $h_1 = e(g, h)$, C_{1i} and C_{2i} are the ciphertexts of message M_{1i} and M_{2i} respectively.

- Secure Addition: Given the ciphertexts of message $M_{11}, M_{12}, \ldots, M_{1v}$, the cloud computes their sum as:

$$C(sum) = \prod_{i=1}^{v} C_i$$

Cloud Secure Computing

- Decryption: without loss of generality, we just demonstrate the decryption of $C(sum)$ as follows. The cloud broadcasts $C(sum)$ to each party. On receiving the ciphertext, each party P_s computes $C(sum)^{q_{1s}}$ and returns the result to the cloud.

  With the results from all the parties, the cloud computes:

$$\prod_{j=1}^{Z} C(sum)^{q_{1s}} = C(sum)^{q_1}$$

  Since $C(sum) = \prod_{i=1}^{v} C_i = \prod_{i=1}^{v} g^{M_i} h^{r_i}$, we have:

$$C(sum)^{q1} = (g^{\sum_{i=1}^{v} M_i} \prod_{i=1}^{v} h^{r_i})^{q_1} = (g^{q_1})^{\sum_{i=1}^{t} M_i}$$

  Note that $h^{q_1} = 1. \sum_{i=1}^{v} M_i$ can be efficiently solved using Pollard's lambda method[1] given g^{q_1}. The encrypted scalar product can be decrypted jointly in the similar way.

[1] Katz, Jonathan, et al. *Handbook of applied cryptography*. CRC press, 1996.

BGN for Large Number

- The Pollard's lambda method is able to decrypt numbers of up to 30-40 bits within a reasonable time slot (e.g., in minutes or hours).
- For larger numbers, let the data holders divide the numbers, if they are large, into several numbers, and the cloud then decrypt the smaller "chunks".

Let $V_A = (A_1, A_2, \ldots, A_k)$ and $V_B = (B_1, B_2, \ldots, B_k)$ be two vectors, where A_i and B_i are $3d$-bit numbers for $1 \le i \le k$. Each number can be represented as

$$A_i = A_{i2} * 2^{2d} + A_{i1} * 2^d + A_{i0},$$

$$B_i = B_{i2} * 2^{2d} + B_{i1} * 2^d + B_{i0}.$$

We can compute the product of $A_i * B_i$ as follows:

$$\begin{aligned} A_i * B_i = {} & 2^{4d}(A_{i2} * B_{i2}) + 2^{3d}(A_{i2} * B_{i1} + A_{i1} * B_{i2}) \\ & + 2^{2d}(A_{i2} * B_{i0} + A_{i0} * B_{i2} + A_{i1} * B_{i1}) \\ & + 2^d(A_{i1} * B_{i0} + A_{i0} * B_{i1}) + A_{i0} * B_{i0}. \end{aligned}$$

Secure Sharing of Sum

Goal- to support consecutive multiplication operations via decryption and re-encryption

- Ciphertext of Sum - $C(S)$
- Each party P_s generate a random number l_s,
- Encrypt l_s and send $C(l_s)$ to cloud
- Cloud obtain $C(sumL = \sum l_s)$, $C(\bar{L}) = C(sumL - S)$, decrypt $C(\bar{L})$ and send $(l_1 - \bar{L})$ to P_1
- Secure share: P_1 has $(l_1 - \bar{L})$, others have l_s

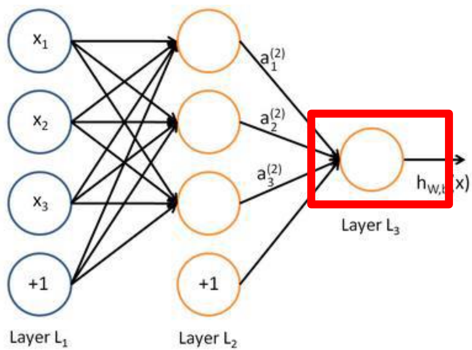
$$C(sumL) = \prod_{s=1}^{Z} C(L_s) = g_1^{L_1+L_2+\cdots+L_Z} h_1^{q_2 r_s},$$

Approximation of Sigmoid Function

- Approximation of activation function using Maclaurin series expansion since BGN encryption does not support exponentiation operation over ciphertext

$$\frac{1}{1+e^{-x}} = \frac{1}{2} + \frac{x}{4} - \frac{x^3}{48} + \frac{x^5}{480} + O(x^6).$$

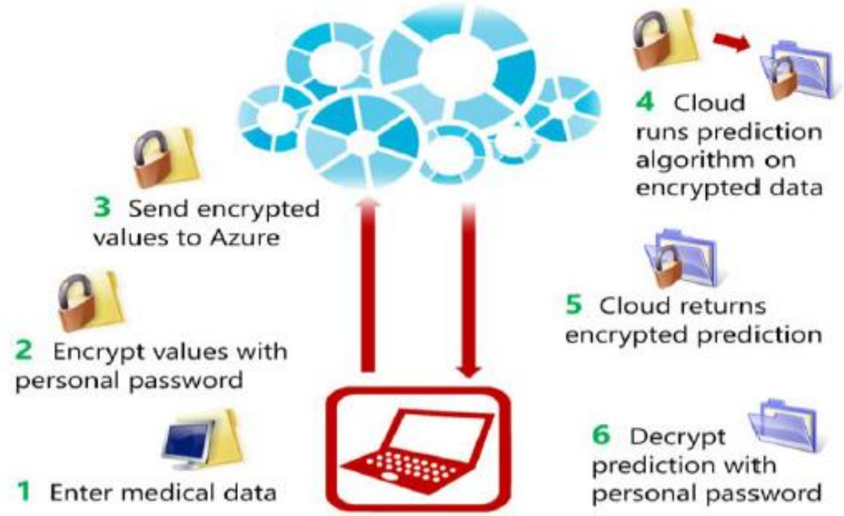
- For $x^k (x = \sum x_s)$, need call k times secure share algorithm



CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

Nathan Dowlin, Ran Gilad-Bachrach, et.al.
16'ICML

Overview



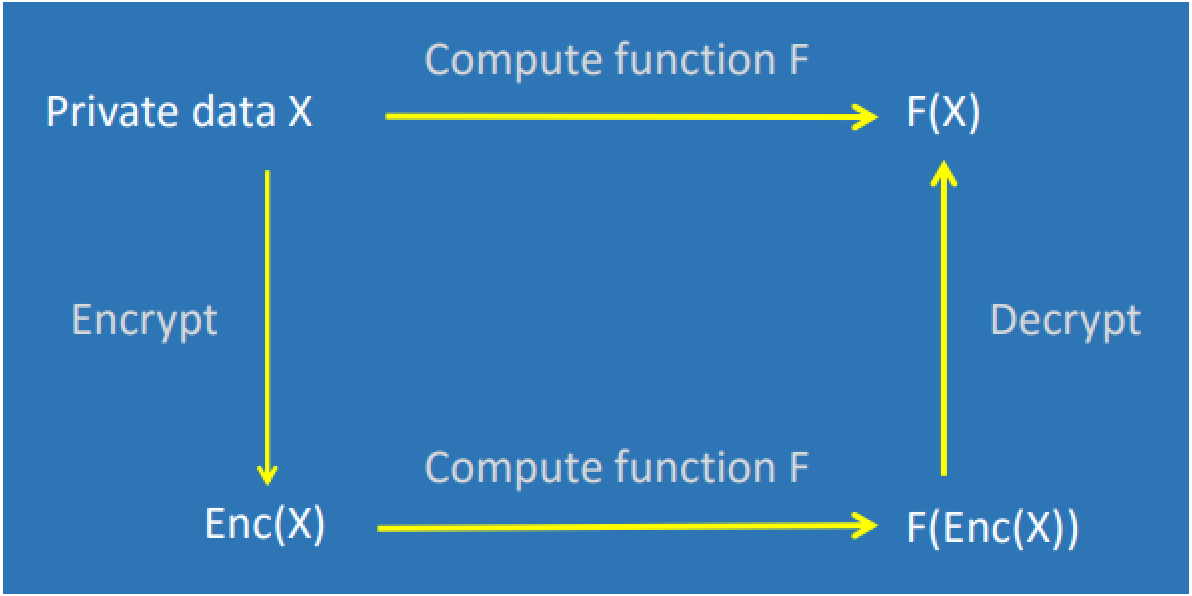
3 Send encrypted values to Azure
2 Encrypt values with personal password
1 Enter medical data
4 Cloud runs prediction algorithm on encrypted data
5 Cloud returns encrypted prediction
6 Decrypt prediction with personal password

Neural Networks

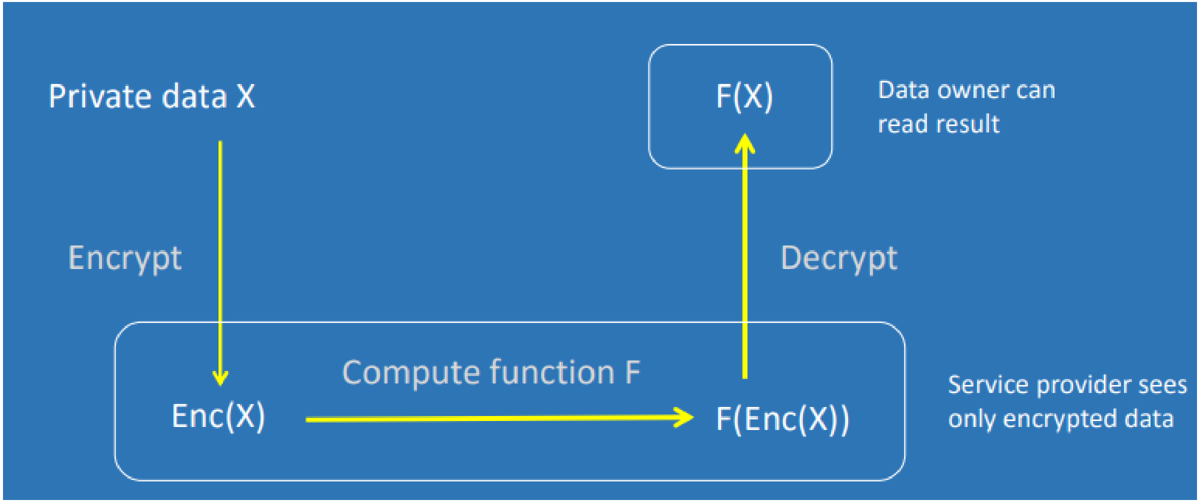
1. Weighted-Sum (convolution layer): a dot product of the weight vector and the vector of values of the feeding layer.
2. Max Pooling: Compute the maximal value of some of the components of the feeding layer.
3. Mean Pooling: Compute the average value of some of the components of the feeding layer.
4. Sigmoid: $\frac{1}{(1+\exp(-z))}$
5. Rectified Linear: max (o ; z).

Homomorphic Encryption

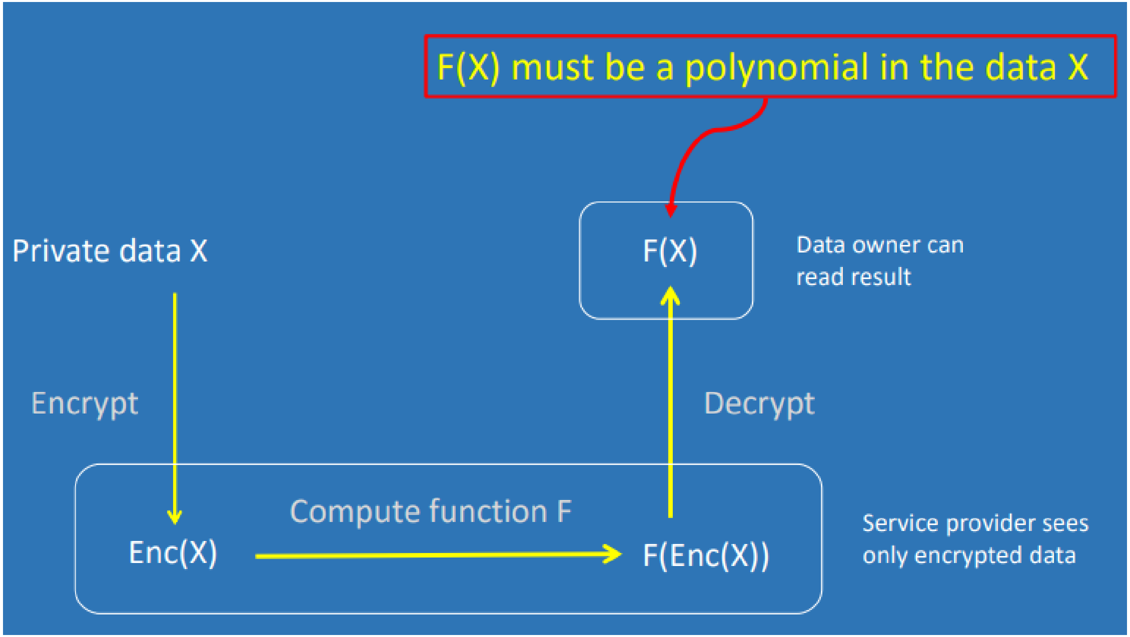
1. F(x) is polynomial function



Homomorphic Encryption



Homomorphic Encryption



Work for Real Number

- Convert the real number to fixed precision numbers, and then convert them into a polynomial with the coefficients
    - Ie. Conversions as encodings (real number to R^n) and decoding (R^n to real numbers).

$$m = \sum_{i=0}^{n-1} a_i X^i, \text{ with } a_i \text{ is integer}$$

YASHE

- Leveled homomorphic encryption
  - 'leveled' refers to the fact that the homomorphic encryption scheme cannot correctly and securely carry out an arbitrary computation; instead, the scheme can only be used to compute functions up to a certain complexity, or level, that is fixed in advance

Encrypt: $c := [\lfloor q/t \rfloor m + e + hs]_q$

Decrypt: $m := \left[\left\lfloor \frac{t}{q} fc \right\rceil\right]_t$.

Message – m Ciphertext -c
Public key – h Private key-f
Parameters -Others

Noise & Correctness
Simple example:

- Enc(m): $\overset{\boldsymbol{b}}{\boxed{m + as + 2e}}\, mod\ q, q$ is public
- Dec(c): $\big((b\ mod\ q) * p\big)\ mod\ 2 = (b - q\lceil {}^{b}/_{q} \rfloor)\ p\ mod\ 2$
  - Bp-asp = $(m + 2e) * p\ mod\ q$
  - Decrypt correctly $e < \frac{q}{2}$

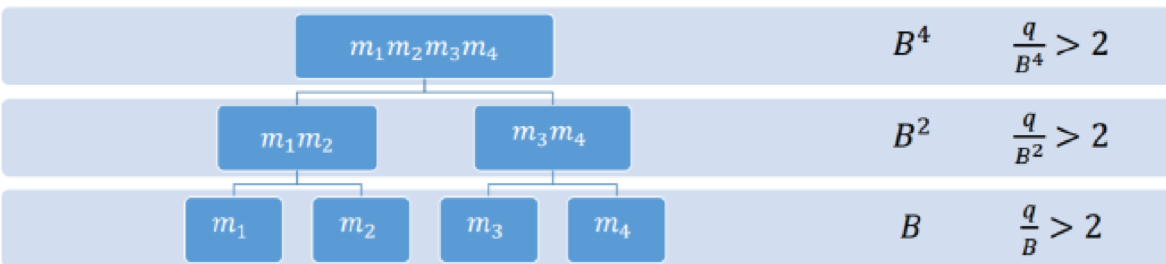
p← private key;

S ←public key; p*s mod q $= 0$

Sample $a \leftarrow R_q$ uniform; e ←error

Noise Growth

- Initial noise: B
- Addition: noise add up, $B \rightarrow 2B$
- Multiplication: noise terms are multiplied, $B \rightarrow B^2$



- $B^2 \rightarrow B^4, B^4 \rightarrow B^8, \ldots, B^{2^{L-1}} \rightarrow B^{2^L}$ (L levels of multiplications)

- For a given t and n, the size of the initial noise and a fixed number of levels of multiplications, we can estimate the size of the resulting noise, and we can then set q large enough to allow for correct decryption

Text Privacy

Two Kinds of Document Privacy Leakage

- Authorship attribution
  - writing style can reveal author identity or other undisclosed personal attributes such as native language, gender or age.

- Text representation
  - User-generated textual data not only can reveal the identity of the user but also may contain individual’s private information.

Visual
Privacy

Privacy in Images and Videos

- Photos or vlogs shared publicly on social platforms may reveal the users'
  - Home location
  - Contact
  - Bank account
  - Family members
  - Other sensitive information

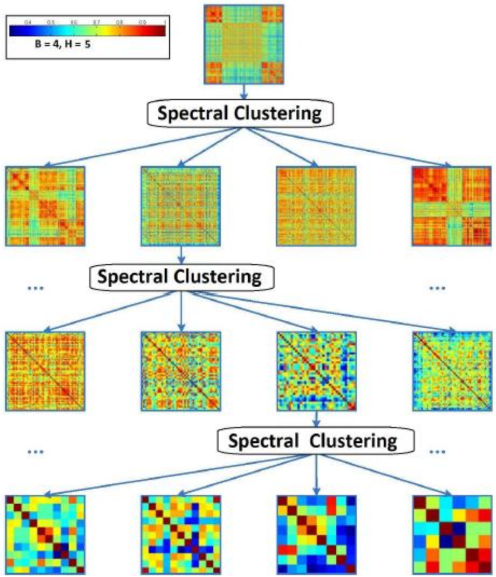
iPrivacy

» Step 1: Deep CNNs

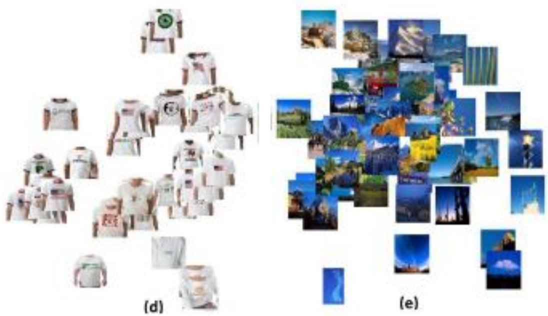
- Semantic image segmentation
- Automatic object-privacy alignment

iPrivacy

» Step 2: A visual tree



- Organize privacy-sensitive object classes hierarchically in a coarse-to-fine fashion
- Each leaf node: privacy-sensitive object class

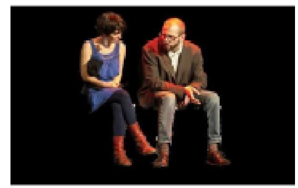


iPrivacy

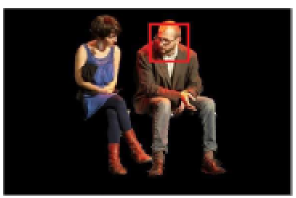
- **Step 3: Tree classifier**
- Example

Original image

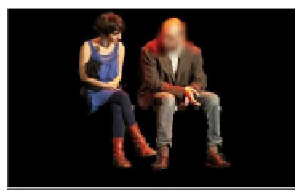
Human region



Detected human objects



Face identification



Face blurring

Shared image

What about Video Privacy?

Audio

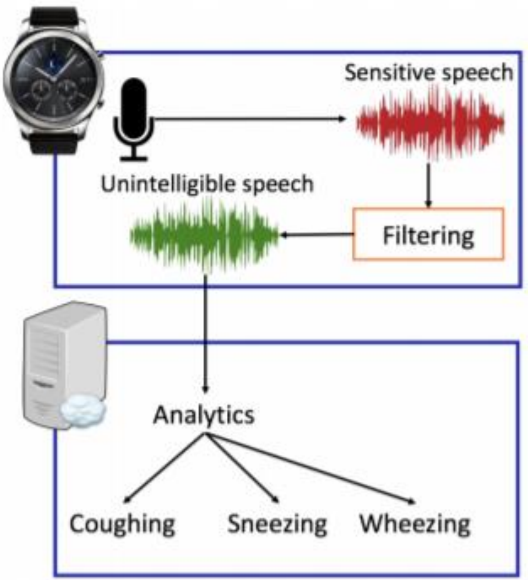
Privacy

Privacy Threats in Audio

- Speech overheard
- Human gestures/behavior leakage
  - through audio sensing signals
- Contact-free monitoring of the health conditions of individuals
  - through breathing patterns

Audio Privacy Protection : Approach 1

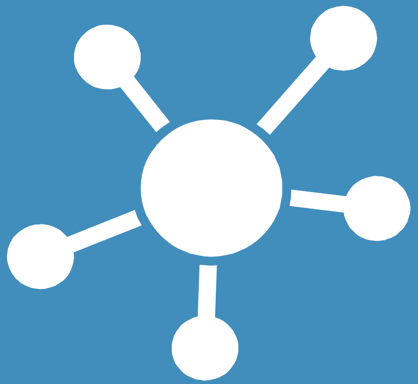
Approach proposed by D. Liaqat et al.



[1] D. Liaqat, E. Nemati, M. Rahman and J. Kuang, "A method for preserving privacy during audio recordings by filtering speech," 2017 IEEE Life Sciences Conference (LSC), Sydney, NSW, 2017, pp. 79-82.

Social Network Privacy

Social networks describe entities (often people) and the relationships between them.



Problem Description

- Model we will model a social network as a simple, undirected graph $G = (V, E)$.
  - Nodes entities, each has a unique name
  - Edges connections between entities

- Goal to remove information pertaining to individual identities, while retaining the topological structure of the graph. We will refer to this de-identified graph as $G' = (V', E')$.

[1] Chen, Bee-Chung & Kifer, Daniel & LeFevre, Kristen & Machanavajjhala, Ashwin. (2009). Privacy-Preserving Data Publishing. Foundations and Trends in Databases. 2. 1-167. 10.1561/1900000008.

Location Privacy

Problem Description

- Cellular service providers and car companies are able to collect location trace data from many mobile users.

- The owners of these repositories may wish to publish, distribute, or sell these data to enable a new set of applications called *location-based services* (LBS).

[1] Chen, Bee-Chung & Kifer, Daniel & LeFevre, Kristen & Machanavajjhala, Ashwin. (2009). Privacy-Preserving Data Publishing. Foundations and Trends in Databases. 2. 1-167. 10.1561/1900000008.

THANKS!

Any questions?

You can find me at:

» zhanglan@ustc.edu.cn