INTERACTING USER TARGET FOR IMAGE SEARCH USING VISUAL QUERY EXPANSION

Sudhakar Murugesan
Lecturer, Department of Information Technology
Valley View University, Ghana, West Africa

Subitha Natesan
M.Tech Department of Computer Science,
Sathyabama University, Chennai, Tamil Nadu, India.

ABSTRACT: *Web-scale image search engines (e.g. Google Image Search, Bing Image Search) mostly rely on surrounding text features. It is difficult for them to interpret users' search intention only by query keywords and this leads to ambiguous and noisy search results which are far from satisfactory. It is important to use visual information in order to solve the ambiguity in text-based image retrieval. In this paper, we propose a novel Internet image search approach. It only requires the user to click on one query image with the minimum effort and images from a pool retrieved by text-based search are re-ranked based on both visual and textual content.*

KEYWORDS: User target, Image search, Query, Text based search

Our key contribution is to capture the users' search intention from this one-click query image in four steps.

- The query image is categorized into one of the predefined adaptive weight categories, which reflect users' search intention at a coarse level. Inside each category, a specific weight schema is used to combine visual features adaptive to this kind of images to better re-rank the text-based search result.
- Based on the visual content of the query image selected by the user and through image clustering, query keywords are expanded to capture user intention.
- Expanded keywords are used to enlarge the image pool to contain more relevant images.
- Expanded keywords are also used to expand the query image to multiple positive visual examples from which new query specific visual and textual similarity metrics are learned to further improve content-based image re-ranking. All these steps are automatic without extra effort from the user. This is critically important for any commercial web-based image search engine, where the user interface has to be extremely simple. Besides this key contribution, a set of visual features which are both effective and efficient in Internet image search are designed. Experimental evaluation shows that our approach significantly improves the precision of top ranked images and also the user experience.

INTRODUCTION

In recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the medical information more efficient. Especially, each patient is promised the full control of her medical records and can share her health data with a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centres, many PHR services are outsourced to or provided by third-party service providers.

The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the other hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates, cloud providers are usually not covered entities. On the other hand, due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviours which may lead to exposure of the PHI.

The goal of patient-centric privacy is often in conflict with scalability in a PHR system. The authorized users may either need to access the PHR for personal use or professional purposes. Examples of the former are family member and friends, while the latter can be medical doctors, pharmacists, and researchers, etc. We refer to the two categories of users as personal and professional users, respectively. The latter has potentially large scale; should each owner herself be directly responsible for managing all the professional users, she will easily be overwhelmed by the key management overhead. In addition, since those users' access requests are generally unpredictable.

The main goal of our framework is to provide secure patient-centric PHR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains (PUDs) and personal domains (PSDs)) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses and medical researchers. In practice, a PUD can be mapped to an independent sector in the society, such as the health care, government or insurance sector. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to PHRs based on access rights assigned by the owner.

REVIEW OF LITERATURE

Alexandra Boldyreva [1], proposed an IBE (Identity Based Encryption) scheme with efficient revocation, whose complexity of key updates is significantly reduced (from linear to logarithmic in the number of users). IBE eliminates the need for a Public Key Infrastructure (PKI). The senders using an IBE do not need to look up the public keys and the corresponding certificates of the receivers. Any setting, PKI- or identity-based, must provide a means to revoke users from the system. Efficient revocation is a well-studied problem in the traditional PKI setting.

Yao Zheng [10], proposed a privacy-preserving PHR system using attribute-based encryption (ABE). In this system, patients can encrypt their PHRs and store them on semi-trusted cloud servers such that servers do not have access to sensitive PHR contexts. Meanwhile patients maintain full control over access to their PHR files, by assigning fine-grained, attribute-based access privileges to selected data users, while different users can have access to different parts of their PHR.This system also provides extra features such as populating PHR from professional electronic health record (EHR) using ABE.

J. Hur [4], proposed data outsourcing systems that requires flexible access control policies. He proposed an access control mechanism using cipher text-policy attribute-based encryption to enforce access control policies with efficient attribute and user revocation capability. He also implements the optimized dynamic re-keying mechanism to reduce the storage cost as well as computational cost. This scheme constructs access table to maintain the access policies and the details of resources. The main aim is used to avoid the duplicate data.

John Bethencourt [8], proposed that, several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. A system for realizing complex access control on encrypted data that calls Cipher text-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential. The methods are conceptually closer to traditional access control methods such as Role-Based Access Control.

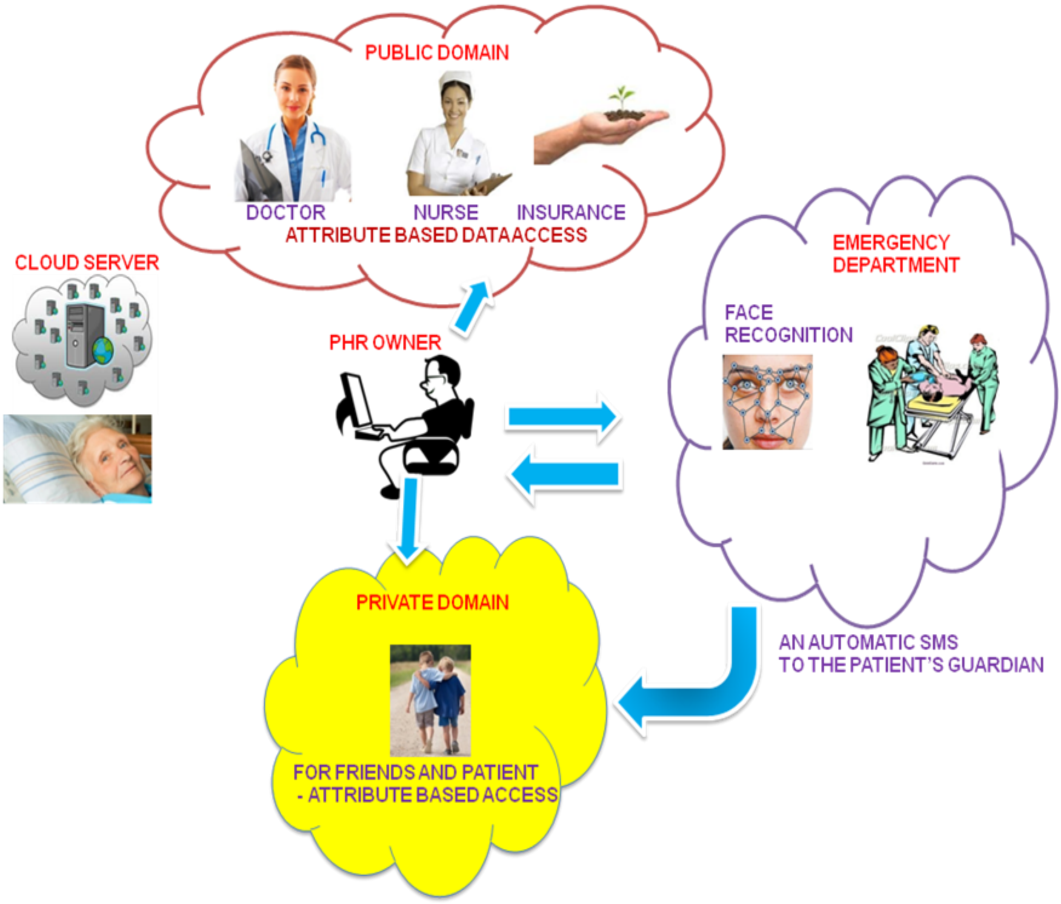
SYSTEM DESIGN

The Design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced. Design elements are required to describe the software features in detail. These design programmers can convert them into code directly without any additional input. Thus the logical system of the product is developed in this stage. This stage also tries to create the architecture of the entire system. This is a benchmark stage since any errors performed till this stage and in this stage can be cleared. Design focuses mainly on high level design (what programs are needed), low level design (how the individual programs are going to work), interface design (how the interfaces going to look like).and data design (what data will be used). The deliverable in this phase is FDD (Functional Design document).

System Design involves identification of classes their relationship as well as their collaboration. The Computer Aided Software Engineering (CASE) tools take advantage of Meta modelling that is helpful only after the construction of the class diagram. Software project is created by both designer and analyst. The analyst creates the user case diagram. The designer creates the class diagram.

ARCHITECTURAL COMPONENTS

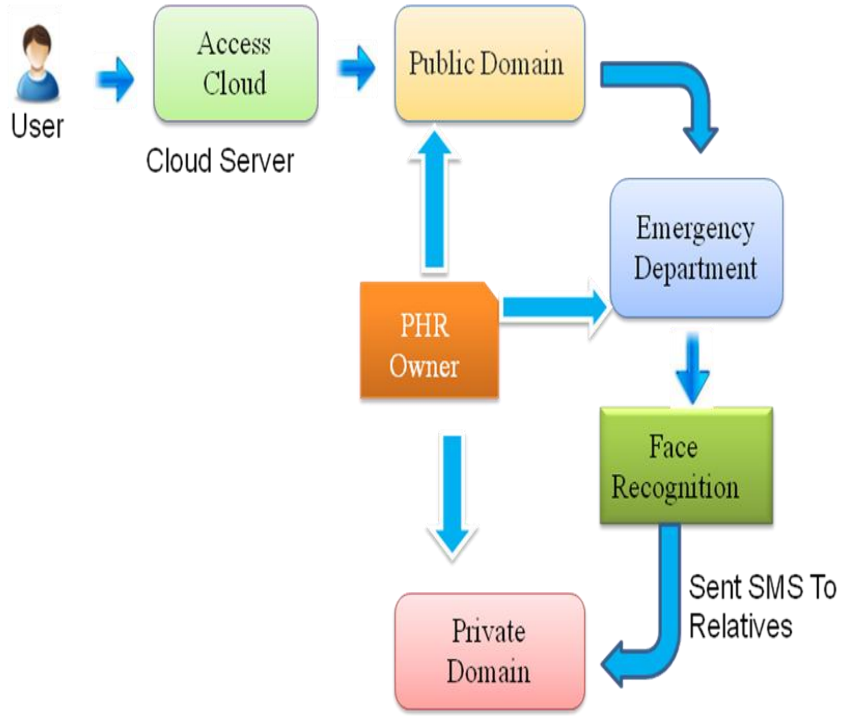
Fig1: System Architecture


PUBLIC DOMAIN
DOCTOR
NURSE
INSURANCE
ATTRIBUTE BASED DATAACCESS
CLOUD SERVER
PHR OWNER
EMERGENCY DEPARTMENT
FACE RECOGNITION
PRIVATE DOMAIN
FOR FRIENDS AND PATIENT - ATTRIBUTE BASED ACCESS
AN AUTOMATIC SMS TO THE PATIENT'S GUARDIAN

The main goal of our framework is to provide secure patient-centric PHR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains (PUDs) and personal domains (PSDs)) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses and medical researchers. PUD can be mapped to an independent sector in the society, such as the health care, government or insurance sector. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to PHRs based on access rights assigned by the owner.

We endeavor to study the patient-centric, secure sharing of PHRs stored on semi-trusted servers, and focus on addressing the complicated and challenging key management issues. To address the key management challenges, we conceptually divide the users in the system into two types of domains, namely public and personal domains. In particular, the majority professional users are managed distributive by attribute authorities in the former, while each owner only needs to manage the keys of a small number of users in her personal domain. In this way, our framework can simultaneously handle different types of PHR sharing applications' requirements, while incurring minimal key management overhead for both owners and users in the system. In addition, the framework enforces write access control, handles dynamic policy updates, and provides break-glass access to PHRs under emergence scenarios.

Fig 2: Data Flow Diagram



The Data Flow diagram is a graphic tool used for expressing system requirements in a graphical form. The DFD also known as the "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that to become program in system design. This DFD can be stated as the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. The DFD consist of series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system. A DFD describes what that data flow in rather than how they are processed. So it does not depend on hardware, software, data structure or file organization

I demonstrate how our frame-work works using a concrete example. Suppose PHR owner Alice is a patient associated with hospital A. After she creates a PHR file F1(labelled as "PHR; medical history; allergy; emergency"), she first encrypts it according to both F1's data labels, and a role-based file access policy. This policy can be decided based on recommended settings by the system, or Alice's own preference. It look like p1:="(profession=physician)∧(specialty=internaledicine)∧(organization=hospital A)". She also sends the break-glass key to the ED. In addition, Alice determines the access rights of users in her PSD, which can be done either on-line or off-line. For example, she may approve her friend Bob's request to access files with labels {personal info}or{medical history}. Her client application will distribute a secret key with the access structure (personal info ∨ medical history) to Bob. When Bob wants to access another file F2 with labels "PHR - medical history - medications", he is able to decryptF2 due to the "medical history" attribute. For another user Charlie who is a physician specializing in internal medicine in hospital B in the PUD, he obtains his secret key from multiple AAs. But he cannot decryptF1, because his role attributes do not satisfyP1. Finally, an emergency room staff, Dorothy who temporarily obtains the break-glass key from ED, can gain access toF1 due to the emergency attribute in that key.

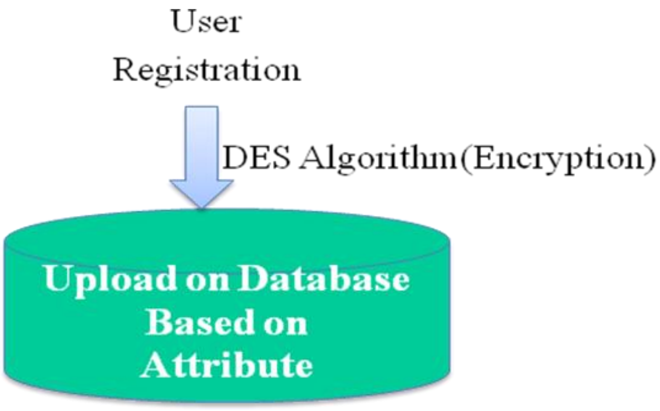
The separation of PSD/PUD and data/role attributes reflects the real-world situation. First, in the PSD, a patient usually only gives personal access of his/her sensitive PHR to selected users, such as family members and close friends, rather than all the friends in the social network. Different PSD users can be assigned different access privileges based on their relationships with the owner. In this way, patients can exert fine-control over the access for each user in their PSDs. Second, by our multi-domain and multi-authority frame-work, each public user only needs to contact AAs in its own PUD who collaboratively generates a secret key for the user, which reduces the workload per AA.

MODULE DESCRIPTION

User registration

In this module, the user is the act of confirming the truth of an attribute of a datum or entity. This module includes the attributes define options like the user details will be uploaded. Option of user to add the data or the data will be uploaded by the PHR owner in the system. We are using DES Algorithm to encrypt the user's details in the system. Registration and activation of new user is performed in the user registration module.

Fig 3: User Registration Module



The system first defines a common universe of data attributes shared by every PSD, such as "basic profile", "medical history", "allergies", and "prescriptions". An emergency attribute is also defined for break-glass access. Each PHR owner's client application generates its corresponding public/master keys. The public keys can be published via user's profile in an online healthcare social-network which could be part of the PHR service; e.g., the Indivo system. There are two ways for distributing secret keys. First, when first using the PHR service, a PHR owner can specify the access privilege of a data reader in her PSD, and let her application generate and distribute corresponding key to the latter, in a way resembling invitations in GoogleDoc. Second, a reader in PSD could obtain the secret key by sending a request (indicating which types of files she wants to access) to the PHR owner and the owner will grant her a subset of requested data types.

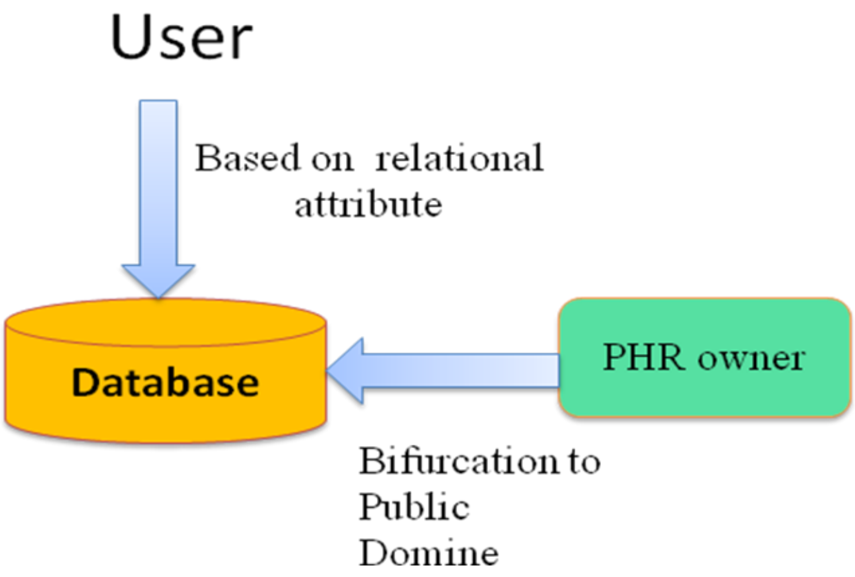
Private Domain Specific Data Point

Specifying the data level of access to the user who are bit of relational in nature to the user were involved in this module. The user is permitted to provide the level of access given to the relations (Based on the relational attribute) is detailed in this module. Each data owner (e.g., patient) is a trusted authority of her own PSD, who uses a encryption system to manage the secret keys and access rights of users in her PSD. Since the users are personally known by the PHR owner, to realize patient-centric access, the owner is at the best position to grant user access privileges on a case-by-case basis. For PSD, data attributes are defined which refer to the intrinsic properties of the PHR data, such as the category of a PHR file. For the purpose of PSD access, each PHR file is labeled with its data attributes, while the key size is only linear with the number of file categories user can access. Since the number of users in a PSD is often small, it reduces the burden for the owner. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties.

Public Domain Specific Data Point

Articulating the data between various public domains is the core idea of this module. In this module, the user or PHR owner will decide the type of data to be shown to the public domain people. A clear bifurcation will be given between the different public domain people.

Fig 4: Public Domain Specific Data Point



In the public domain, we use multi-authority ABE (MA-ABE) to improve the security and avoid key escrow problem. Each attribute authority (AA) in it governs a disjoint subset of user role attributes, while none of them alone is able to control the security of the whole system. We propose mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access policies during file encryption. In the personal domain, owners directly assign access privileges for personal users and encrypt A PHR file under its data attributes

CONCLUSION

A novel framework of secure sharing of personal health records in cloud computing is proposed. Considering partially trustworthy cloud servers, we argue that to fully realize the patient-centric concept, patients shall have complete control of their own privacy through encrypting their PHR files to allow fine-grained access. The framework addresses the unique challenges brought by multiple PHR owners and users, in that we greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. We utilize ABE to encrypt the PHR data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications and affiliations

REFERENCES

1. A. Boldyreva, V. Goyal, and V. Kumar, “Identity-based encryption with efficient revocation,” inACM CCS, ser. CCS ’08, 2008, pp.417–426.
2. C. Dong, G. Russello, and N. Dulay, “Shared and searchable encrypted data for untrusted servers,” in Journal of Computer Security, 2010.
3. H. L¨ ohr, A.-R. Sadeghi, and M. Winandy, “Securing the e-health cloud,” inProceedings of the 1st ACM International Health Informatics Symposium, ser. IHI ’10, 2010, pp. 220–229
4. J. Hur and D. K. Noh, “Attribute-based access control with effi-cient revocation in data outsourcing systems,”IEEE Transactions on Parallel and Distributed Systems, vol. 99, no. PrePrints, 2010.
5. L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, “Ciphertext-policy attribute-based threshold decryption with flex-ible delegation and revocation of user attributes,” 2009.
6. M. Li, S. Yu, K. Ren, and W. Lou, “Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings,” inSecureComm’10, Sept. 2010, pp. 89–106
7. S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” inIEEE INFOCOM’10, 2010.
8. S. Yu, C. Wang, K. Ren, and W. Lou, “Attribute based data sharing with attribute revocation,” inASIACCS’10, 2010.
9. X. Liang, R. Lu, X. Lin, and X. S. Shen, “Patient self-controllable access policy on phi in ehealthcare systems,” inAHIC 2010, 2010.
10. Yao Zheng “Privacy-preserving personal health record system using attribute-based encryption,” Master’s thesis, WORCESTER POLY-TECHNIC INSTITUTE, 2011.