2011 31st International Conference on Distributed Computing Systems

HCPP: Cryptography Based Secure EHR System for Patient Privacy and Emergency Healthcare

Jinyuan Sun*, Xiaoyan Zhu†, Chi Zhang‡, and Yuguang Fang‡
* University of Tennessee, Knoxville, TN 37996, email: jysun@eecs.utk.edu
† Xidian University, Xi'an, China, email: xy.ok@hotmail.com
‡ University of Florida, Gainesville, FL 32611, email: {zhangchi@, fang@ece.}ufl.edu

***Abstract*—Privacy concern is arguably the major barrier that hinders the deployment of electronic health record (EHR) systems which are considered more efficient, less error-prone, and of higher availability compared to traditional paper record systems. Patients are unwilling to accept the EHR system unless their protected health information (PHI) containing highly confidential data is guaranteed proper use and disclosure, which cannot be easily achieved without patients' control over their own PHI. However, cautions must be taken to handle emergencies in which the patient may be physically incompetent to retrieve the controlled PHI for emergency treatment. In this paper, we propose a secure EHR system, HCPP (Healthcare system for Patient Privacy), based on cryptographic constructions and existing wireless network infrastructures, to provide privacy protection to patients under any circumstances while enabling timely PHI retrieval for life-saving treatment in emergency situations. Furthermore, our HCPP system restricts PHI access to authorized (not arbitrary) physicians, who can be traced and held accountable if the accessed PHI is found improperly disclosed. Last but not least, HCPP leverages wireless network access to support efficient and private storage/retrieval of PHI, which underlies a secure and feasible EHR system.**

***Index Terms*—Privacy, Security, Emergency, EHR, Wireless Networks.**

I. Introduction

The development and management of medical systems concern all of us since we will inevitably be the users of these systems. The major concern is clearly the privacy of patients and their medical records which reveal highly confidential personal information such as disease history and undergoing treatment. There are good reasons for keeping the records private and limiting the access to only minimum necessary information: an employer may decide not to hire someone with psychological issues, an insurance company may refuse to provide life insurance when knowing the disease history of a patient, a person with certain types of disease may be discriminated by the healthcare provider, and so on. However, fundamental developments of heath care systems have threatened the confidentiality of medical records and patient privacy [1], one of which is the exponential increase in the use of computers and automated information systems for health record information. It is now common to see physicians use computers (connected to a network) to store and retrieve patients' electronic health records (EHRs).

EHR systems are used in place of paper systems to increase physician efficiency, reduce costs (e.g., storage) and medical errors, improve data availability and sharing, etc. An exemplary successful implementation of EHR system in the United States is the Veterans Administration healthcare system, with over 155 hospitals and 800 clinics. It is one of the largest integrated healthcare information systems worldwide and has been using a single EHR system for years. Despite all the promising factors, EHR systems are not adopted by the majority of healthcare systems. Statistical results of the actual adoption rate of EHR in US medical systems can be found in Section 4 of [2] and the references therein. Among all the barriers to the implementation of EHR systems, privacy and security concerns on patients' medical records are arguably most dominating. Records stored in a central server and exchanged over the Internet are subject to theft [3] and security breaches. The Health Insurance Portability and Accountability Act (HIPAA) in the US was established to regulate EHR related operations. In addition to governmental regulations, standardization and an overall strategy are needed to ensure that privacy protections would be built into computer networks linking insurers, doctors, hospitals and other healthcare providers [4]. The implementation of the standardization or strategy will undoubtedly be relying on technical details, which are rarely studied in the research realm and open numerous research opportunities.

A. Related Work

The majority of works on privacy protection in healthcare systems still concentrate on the framework design or solution proposals without technical realization [5]–[9]. These works include the demonstration of the significance of privacy for EHR systems, the authentication based on existing wireless infrastructure, the role-based approach for access restrictions, etc. As the need for technical details, specifically, the cryptographic realization of privacy and security in healthcare systems becomes more clear and stringent, a few recent works followed this line of research. Lee and Lee [10] proposed a cryptographic key management solution for privacy and security regulations regarding patients' PHI. Patients have control over their PHI and are able to restrict access to it. When the physician needs to review the PHI for treatment, he has to obtain agreement or consent from patients who will use the proper keys stored on a smart card to decrypt the PHI ciphertexts. The authors then proposed a consent exception solution for emergencies, where a trusted server possesses all secret keys of the patient and hence can retrieve the PHI plaintexts upon emergency. Although technically correct, the proposed scheme is unreasonable since the trusted server is able to access the patients' PHI at any time. As a result, PHI privacy is not fully guaranteed which is unacceptable for extremely sensitive information like PHI. Furthermore, the authors did not address the issues related to storing and retrieving PHI, which can be intricate given the privacy requirements. The work of Tan *et al.* [11] is a technical realization of the role-based approach proposed in [7], though in a limited healthcare setting where body sensor network is employed. The work mainly deals with emergency care scenario, in which privacy concerns and access

1063-6927/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCS.2011.83

restrictions are the focus. In spite of specifying the algorithms for storing and retrieving healthcare records, the scheme in fact failed to achieve privacy protection in that the storage site will learn the ownership of the encrypted records (i.e., which records are from which patient) in order to return the desired records to the querying doctor. Such leakage will compromise patients' privacy by violating the unlinkability requirement. There are also many other works on the secure operations (e.g., delegation, access control, revocation) of healthcare systems in normal cases [12] and on the authentication in body sensor networks [13].

B. Our Contributions

The work of [10] and [11] has some relevance to our work, in which we strive to design protocols for a secure healthcare system leveraging cryptographic tools. The proposed system provides both full privacy for patients without escrow (e.g., the trusted server in [10]), and the capability of handling emergency situations, which are intrinsically related and somehow contradictory. On the one hand, full privacy means even when the patient is incapable of authorizing the access to his PHI during emergencies, no one should be able to obtain the secrets for retrieving and decrypting the PHI. On the other hand, there must be a way to retrieve and decrypt the PHI (as if the patient is conscious to do so) for life-saving purposes in emergencies. In addition, the storage and retrieval of PHI in a secure and private manner underlie the healthcare system and must be carefully coped with. We summarize our contributions as follows:

1. We propose an EHR system, HCPP (Healthcare system for Patient Privacy), that enables patients to efficiently store and retrieve their PHI in a secure and private manner even with a public server, such that only the patient can learn the content of his PHI. Moreover, no one is able to link any (encrypted) PHI files to a particular patient. However, PHI must be disclosed to the physician at the time of treatment or related healthcare operations. Our HCPP is able to trace the physician and hold him accountable if the PHI is later found to be disclosed illegally (e.g., by noticing that the insurance companies refuse to provide life insurance).
2. When the patient is physically incapable of retrieving his PHI (to be mentioned interchangeably with emergency), HCPP provides backup mechanisms that allow the physician to obtain the patient's relevant PHI without compromising the secrets associated with the PHI retrieval, so that the physician is unable to view the PHI other than that was given.
3. HCPP ensures that only physicians with certain access rights (e.g., the emergency caregiver on duty, the family doctor) can access the PHI. Authentication alone would not suffice since a legitimate physician of the same hospital (or clinic) may only be eligible to serve certain types of patient and review some portion of the PHI.
4. We show that HCPP meets the security goals of confidentiality, data integrity, availability, access control, accountability, and fail-open, in addition to the privacy goal mentioned above. We analyze the efficiency of HCPP and discuss possible attacks and countermeasures, which demonstrates the feasibility of our proposed system.

II. Preliminaries

A. IBC from Bilinear Pairings

Identity-based cryptography (IBC) allows the public key of an entity to be derived from its public identity information such as name, email address, etc. Boneh and Franklin [14] introduced the first functional and efficient ID-based encryption scheme based on bilinear pairings on elliptic curves. Specifically, let G_1 and G_2 be an additive group and a multiplicative group, respectively, of the same prime order q. Discrete logarithm problem (DLP) is assumed to be hard in both G_1 and G_2. Let P denote a random generator of G_1 and $e : G_1 \times G_1 \rightarrow G_2$ denote a bilinear map constructed by modified Weil or Tate pairing with the following properties:

1. Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, $\forall P, Q \in G_1$ and $\forall a, b \in Z_q^*$.
2. Non-degenerate: $\exists P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3. Computable: there exists an efficient algorithm to compute $e(P, Q), \forall P, Q \in G_1$.

IBC schemes are used for encryption, authentication, and deriving shared keys in our HCPP protocols (cf. Section IV). Compared to the conventional PKI (public key infrastructure), IBC infrastructure avoids the use of certificates for public key verification and the exchange of public keys (and associated certificates), greatly improving the computation and communication efficiency.

B. Searchable Symmetric Encryption

Searchable symmetric encryption (SSE) is used when an owner of private data wishes to store his data in a remote server that is not trusted. The privacy of the data should be maintained such that the server cannot learn the content of the data, which is achieved by the owner encrypting the entire data before storage. Moreover, the server must be able to search over the encrypted data for a keyword of the owner's choice and return all (encrypted) documents containing this keyword to the owner. Early work on SSE was proposed by Ostrovsky [15], and Goldreich and Ostrovsky [16]. Curtmola *et al.* [17] improved the security definitions of SSE and proposed efficient schemes for secure SSEs. Two important building blocks of SSE are pseudonym-random function (PRF) and pseudo-random permutation (PRP). A PRF is a polynomial-time computable function which emulates a random oracle such that no efficient algorithm can distinguish (with significant advantage) between a PRF and a random oracle. Specifically, let m and ℓ be polynomials, and k be the security parameter. Let $\Re_k$ be the set of all functions: $\{0,1\}^{m(k)} \rightarrow \{0,1\}^{\ell(k)}$. Let $F_k = \{f_s\}_{s \in \{0,1\}^k}$, $F_k \subseteq \Re_k$, be a set of functions indexed by seeds s. We say F_k is a PRF family if the following properties are satisfied:

1. Efficiency: Given x and seed s, $f_s(x)$ is computable in deterministic polynomial time.
2. Pseudorandomness: For all non-uniform oracle PPT (probabilistic polynomial time) adversaries A, $\Pr_{s \leftarrow \{0,1\}^k}[A^{f_s} = 1] - \Pr_{R \leftarrow \Re_k}[A^R = 1]$ is negligible in k, where $x \leftarrow y$ denotes x is randomly selected from y.

A PRP is a PRF that is also a permutation, that is, for every seed s and every n, f_s is a permutation when restricted to n-bit strings.

The private storage and retrieval of patients' (encrypted) PHI in our secure EHR system is constructed based on [17] to guarantee patient privacy. The capability of searching over encrypted data is provided by additional data structures, a secure index for the entire data collection, and a trapdoor for each keyword being searched. These data structures are sent to the server together with the encrypted data collection. Note that we apply the non-adaptive SSE construction in [17] to our protocols for demonstration. The adaptive SSE construction [17] which features a more robust security notion can be applied instead without modifying other parts of the protocols.

C. Searchable Public-Key Encryption

Public key encryption with keyword search (PEKS), or simply searchable public-key encryption, allows an email server to tell if a given keyword is present in emails destined to the receiver without learning anything else about the encrypted emails. Unlike SSE, the data (encrypted with the receiver's public key) are stored in the remote server by the sender, and will be decrypted and used by the receiver. The receiver generates trapdoors for keywords of his choice and sends the trapdoors to the server. The server searches over the encrypted data from the sender and only returns data containing particular keywords to the receiver. The encryption and decryption in PEKS are performed by different parties, where public-key encryption should be employed.

Boneh $et\ al.$ [18] first established the security definitions of PEKS and provided a construction based on the identity-based encryption (IBE) [19]. However, Abdalla $et\ al.$ [20] later found that there exist IBE schemes such that the PEKS derived using the construction of [18] is not computationally consistent (note that security and consistency are the two conditions every cryptographic primitive should satisfy). The authors in [20] then proposed a new transformation from an IBE to a PEKS which is both secure and computationally consistent. The new PEKS construction is similar to the PEKS in [18] except that [20] encrypts a random message R while [18] always uses 0^k as the message encrypted. We will use [18] for demonstration in our emergency healthcare scenario, where the monitored health information (MHI) obtained from high-risk patients is encrypted at the patient side and retrieved later by authorized physicians. Note that the construction in [20] can be adopted instead of that in [18] as a building block of our HCPP system.

III. System Model

A. Overview

We overview our approach and briefly introduce the functionalities of HCPP before diving into technical details. According to the HIPAA privacy regulation rules [1], patients have rights to control the use and disclosure of their PHI, and file complaints regarding illegal disclosures. HCPP thus allows patients to encrypt the PHI and later decrypt them upon legitimate requests from designated physicians. The storage and retrieval of PHI are by no means trivial in that for one thing, HCPP supports PHI storage on a public server while preserving PHI privacy. For another, HCPP facilitates efficient retrieval which indicates the storage server must be able to easily search over the encrypted PHI for a chosen keyword. In emergencies, HIPAA privacy rules permit access to the PHI without the patient's consent because the patient is physically unavailable. It is, however, under the assumption that the physicians make good faith efforts to keep the PHI for legitimate use only. Otherwise, a privacy breach will be introduced. HCPP addresses this concern by exploiting the patient's trust relationship to other entities that are truly trustworthy (e.g., family members he trusts, his own device). These entities will perform PHI retrieval on behalf of the patient. Complication arises when this entity is a patient-owned device due to the lack of subjective judgments on the access right of a physician. HCPP therefore provides a mechanism to trace the physician(s) from whom the PHI is (likely) leaked. The patient can consequently file complaints against the physician(s) after the emergency is resolved.

Entities and Definitions: The following entities are involved in HCPP system: *patient*, *physician*, *S-server* (storage server), *family*, *P-device* (private device), *A-server* (authentication server).

- *Patient* is the user of HCPP system, and is referred to as the combination of a person and his computing facilities (personal desktop, or any wireless-enabled portable devices) for performing necessary computations in PHI storage/retrieval (cf. Section IV), unless otherwise specified. We use cell phone as the wireless portable device in our system in that cell phone is owned by a majority of people [6]. Using cell phone as an essential tool for protocol design is deemed feasible and widely proposed for various systems and applications [6], [21]–[23]. Other devices such as PDA and laptop, can certainly be alternatives of cell phone.
- *Physician* denotes healthcare providers in general, including doctors, assistants, nurses, pharmacists, and any other persons licensed to provide healthcare services. Similar to *patient*, *physician* refers to a person and his work station.
- *S-server* is provided by each hospital/clinic to store the patient's PHI. It can be considered as a public server and is not trusted by patients. The hospital/clinic in general can be assumed honest-but-curious and will not maliciously delete patients' PHI for gaining nothing.
- *Family* represents any persons of the patient's trust including parents, children, spouse, relatives, or in the rare case close friends, who will possess the secrets for retrieving the patient's PHI and are assumed extremely unlikely to compromise the patient's privacy in any case.
- *P-device* refers to electronic devices the patient owns, such as smartphone, PDA, and some wearable devices like the cloaker [8]. P-device is subject to loss and theft, and the subsequent compromises. In general, P-device is different from the computing facilities of *patient* in that it should have the capability for more advanced tasks (e.g., efficient cryptographic computations, wireless networks access, etc) than monitoring and data collection. Moreover, P-device must be pervasive for patients who are highly risky to encounter emergencies.
- *A-server* is a trusted server run by the government (e.g., a state department of health, the US department of health and human services) or its local offices.

In addition to the entities, two important types of information need to be defined.

- PHI, the protected health information, denotes individually identifiable health information in any form (i.e., electronic, paper or oral). PHI also refers to information with respect to which there is a reasonable basis to believe the information

can be used to identify the individual [24]. We are interested in the electronic PHI in our EHR system, HCPP. Strictly speaking, an EHR or health record in general contains PHI and de-identified information (e.g., medical data in the health record that contains no identifiers defined by PHI). In our HCPP system, the patient will encrypt both PHI and de-identified information as a complete piece of record, in order to easily maneuver the storage/retrieval for common-case treatment and emergencies. For the ease of description, we use the name PHI to denote this complete piece of health record.

- MHI, the monitored health information, is the data collected by the monitoring equipments (e.g., sensors) worn or carried by high-risk patients who are considered more likely to experience emergencies. Unlike PHI, MHI is not available for all patients but those under monitoring.

B. System Architecture

Consider the healthcare scenario in our HCPP system shown in Fig. 1, where all links are bidirectional and the bracketed numbers indicate major events or exchanged messages. In general, the

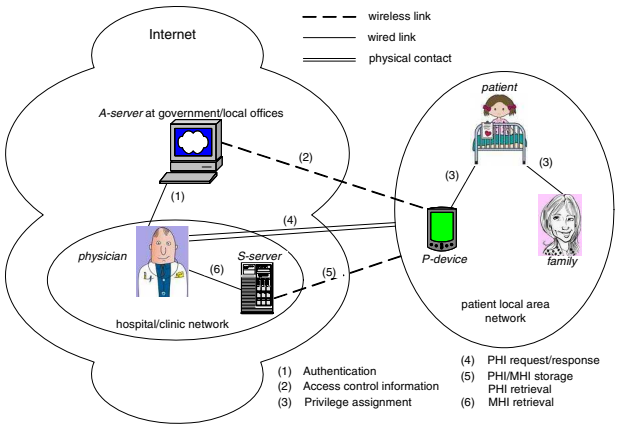

Fig. 1. System Architecture of HCPP.

physician has only physical contacts with all entities in the patient local area network (LAN), denoted by a double solid line from the physician to the patient LAN. Specifically, the physician orally communicates with the patient and family, in common-case treatment and emergencies, respectively. Contacts with P-device, on the other hand, is through the physician physically operating P-device, in emergencies only. Similarly, S-server interacts with all entities in the patient LAN mainly via wireless links, for PHI storage and retrieval. Note that PHI storage is carried out only between S-server and the patient using the patient's home PC. PHI retrieval can be performed by the family and P-device in emergencies, and by the patient in common-case treatment using his cell phone. The internal links of the hospital/clinic network and the patient LAN are often high-speed wired links. The patient interacts with his family and P-device to assign privilege (i.e., secret keys) that will be used for retrieving the patient's PHI in emergencies. The physician and S-server will be engaged in the patient's MHI retrieval, if such information is needed by the physician in emergencies for more effective treatment. The A-server represents the authentication servers of a state or the federal government. The A-server is used for authenticating physicians to determine their eligibility for accessing a particular patient's PHI in emergencies. P-device will then be informed by the A-server regarding whether the authenticating physician has the access right to operate P-device.

C. Security Requirements

HCPP satisfies the following security requirements:

Privacy: HCPP achieves privacy if patients' PHI can *only* be accessed by authorized physicians for legitimate reasons (i.e., treatment, payment, healthcare operations [1]), and no one except the family and P-device can link the stored PHI files to a particular patient.

Fail-open: We say that HCPP system is fail-open if the system provides backup mechanisms to successfully retrieve patients' PHI in the case of emergency while preserves the above privacy properties.

Access Control (Authorization): HCPP realizes access control if no physicians other than authorized (e.g., emergency caregiver on duty) gain access to the patient's PHI.

Accountability: HCPP meets the accountability goal if the physician who disclosed the patient's PHI other than legitimate reasons is traceable and held responsible in the case of emergency. We implicitly assume that when the patient is physically competent to retrieve the PHI, he will know the source of the PHI leakage by recalling which physician(s) recently treated him.

Data Integrity: HCPP guarantees that the stored patients' PHI is not modified except by authorized physicians upon patients' consent or requests. Additionally, protocol messages exchanged between communicating parties are not to be modified by any malicious parties.

Confidentiality: Confidentiality requires that the PHI content is not learned by any eavesdroppers. It is implied by the privacy requirement that all patients' PHI is encrypted. Furthermore, message exchanges involving secret information are subject to confidentiality requirement as well.

Availability: The availability requirement states that the authorized physician must be able to obtain the patient's PHI stored at other hospitals where the patient previously visited.

IV. Construction of the HCPP System

A. System Setup

System setup is executed by A-servers and patients to initialize the security domain for protocol message exchanges, and the HCPP system for health information storage/retrieval, respectively. Each A-server of a state performs IBC domain initialization by inputting security parameter ξ into parameter generator PG, which outputs public domain parameters (q, G_1, G_2, e, P) where P is a generator of G_1. The state A-server randomly selects a master secret $s_0 \in_R Z_q^*$ and computes the domain public key $P_{pub} = s_0 P$. Each physician i or S-server with identity ID_i in the A-server's domain is assigned a public/private key pair PK_i/Γ_i: $PK_i = H_1(ID_i)$, $\Gamma_i = s_0 \cdot PK_i$, where $H_1 : \{0,1\}^* \rightarrow G_1$ is a published hash function. Moreover, the A-server assigns a pool of temporary public/private key pairs to the hospital for patients' use (cf. Section IV. B). The state A-server proceeds to participate in the initialization of an HIBC (hierarchical IBC) domain by having the A-server of the federal government act as the root PKG (public key generator). The federal A-server is at the same time an entity at level 1 of the hierarchical tree. In general, the lower-level

(level j) setup is performed by the parent PA at level j-1. PA computes $K_j = H_1(ID_1, ..., ID_j)$ where $(ID_1, ..., ID_j)$ denotes the collection of all identities on the path connecting a child at level j to the level 1 ancestor. PA further computes a private key for each child at level j as $\psi_j = \psi_{j-1} + s_{j-1}K_j$ where s_{j-1} is PA's randomly chosen secret, and distributes $\{Q_l : 1 \leq l < j\}$ to each child where $Q_l = s_l P$. Note that all state A-servers, and hospitals/clinics in our system are at levels 2, and 3, respectively. Specifically, level 3 includes all affiliated physicians and S-servers of the hospitals/clinics. The root PKG publishes two hash functions $H_2 : KW \rightarrow G_1$ and $H_3 : G_2 \rightarrow Z_q^*$, where KW is the set of all keywords used in PEKS (cf. Section II. C). In the following context, the A-server is referred to as the state A-server unless otherwise specified.

The patient creates a keyword index KI for SSE (cf. Section II. B) recording the association of all keywords and their resulting files, before encrypting the PHI files. The keyword index is for the patient's own reference to facilitate future retrievals and is stored in the personal computer or as a paper document. Let k, γ be security parameters for generating SSE secret keys. The patient runs SSE key generation algorithm (possibly on a home PC) to generate secret keys $a, b, c, d \in_R \{0,1\}^k$ and output $S = \{a, b, c, d, 1^\gamma\}$. He then selects three PRPs $\Re$, ϕ, and θ, a PRF f and a semantically secure symmetric key encryption E as:

$$\Re : \{0,1\}^k \times \{0,1\}^\beta \rightarrow \{0,1\}^\beta$$
$$\phi : \{0,1\}^k \times \{0,1\}^{log_2^\alpha} \rightarrow \{0,1\}^{log_2 \alpha}$$
$$\theta : \{0,1\}^k \times \{0,1\}^{\beta+\gamma+log_2^\alpha} \rightarrow \{0,1\}^{\beta+\gamma+log_2\alpha}$$
$$f : \{0,1\}^k \times \{0,1\}^\beta \rightarrow \{0,1\}^{\gamma+log_2\alpha}$$
$$E : \{0,1\}^\gamma \times \{0,1\}^\tau \rightarrow \{0,1\}^\tau$$

where β is the parameter determining the size of the virtual address in the lookup table T (cf. Section IV. B), α is the total size of the plaintext file collection in bytes, and τ is the size of a node in the array A (cf. Section IV. B). The PRP θ is distributed to *family*, *P-device*, and *S-server*. The secret keys b, c, and functions $\Re$, f, together with the keyword index KI, and a dictionary recording all possible keywords, are stored in the patient's cell phone for future PHI retrieval. In addition, the patient employs a well-established symmetric-key encryption E' and identity-based encryption [19] IBE to encrypt the PHI file collection, and the MHI if available (cf. Section IV. E), respectively. The secret key s used in E' is stored in the patient's cell phone. Lastly, the patient chooses a broadcast encryption BE for the *privilege assignment* protocol presented next, and sends d, $BE_U(d)$ to U and *S-server* where $U = \{family, P\text{-}device\}$ is the set of entities with searching privilege.

B. Private PHI Storage

This protocol provides privacy protection for patient during the PHI storage and future retrieval, and is executed by the patient whenever the PHI is created, updated or modified (e.g., after diagnosis or tests). Note that we let the patient break the PHI into files for different categories of health information (e.g., allergy lists, drug history, X-ray data, surgeries, etc). Each category can also consist of multiple files. The protocol begins with the construction of the secure index SI, including two key data structures that will later enable the S-server to search for the desired (encrypted) files.

An array A is used to store a collection of linked lists. A linked list L_i is a data structure of three fields: file identifier fid, secret key λ for encrypting the next node in L_i, and pointer pr which is the output of ϕ pointing to the address (in A) of the next node in L_i. One linked list is created for each set of file identifiers $\Lambda(kw)$ containing keyword kw from the entire encrypted file collection Λ. As a result, a node in A is of the form $(fid \parallel \lambda \parallel pr)$ where $\parallel$ denotes the concatenation. Note that A may contain an fid in more than one node since searching under different keywords can result in a same file. Nodes are scrambled and stored in A so that one cannot tell which set of files contain the same keyword (i.e., in the same linked list). Another critical data structure in the secure index SI is the lookup table T with entries $\langle VirtualAddress, 1^{st}Node \rangle$, where the former field is the address used to locate the latter field in T. The $1^{st}Node$ field records the encrypted first node in each linked list L for the S-server to locate L. Entries in T are also scrambled as in A using random permutation. The construction of SI=(A,T) is shown in Fig. 2, where $F(kw_i)$ denotes the set of file identifiers containing keyword kw_i from the entire plaintext file collection F, $addr_{i,1}$ and $A[x]$ denote the address of $A(L_{i,1})$ and the element stored at address x in A, respectively. The protocol then stores SI

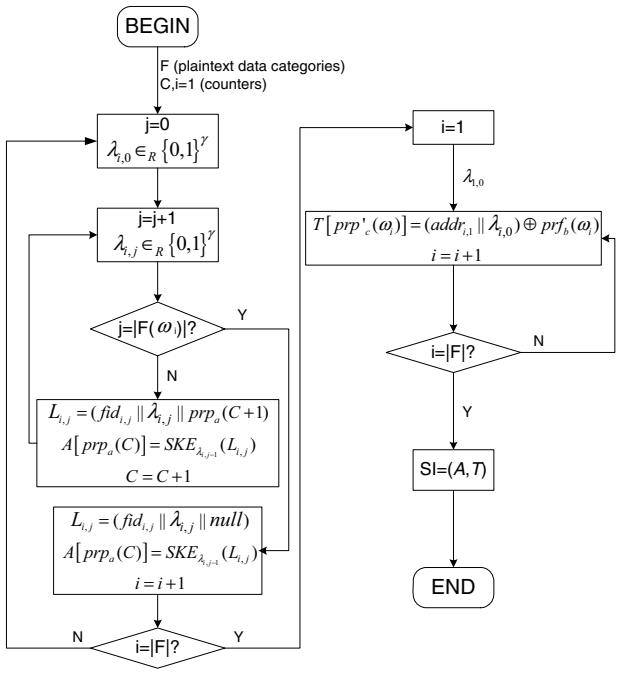

Fig. 2. Construction of the Secure Index.

and the encrypted file collection $\Lambda = E'_s(F)$ in the S-server. From the visited hospital, the patient can obtain the URL link to the S-server, and a temporary public/private key pair, based on which the patient can generate a new valid key pair TP_p/Γ_p (cf. pseudonym self-generation in [25]), so that *S-server* and any other malicious parties are unable to link an activity to a patient by the original key pair assigned by the hospital. The patient then uploads the PHI (i.e., SI and Λ) after building the SI:

$patient \rightarrow S\text{–}server$: TP_p, SI, Λ, t_1,
$HMAC_\nu(TP_p \parallel SI \parallel \Lambda \parallel t_1)$,

where t_1 is the current system time and is included to prevent replay attack [26], $HMAC_\nu$ is a keyed-hash message authentication code for ensuring message integrity, and $\nu = e(\Gamma_p, ID_{S-server}) = e(TP_p, \Gamma_{S-server})$ is the shared key between the patient and S-server computed locally by both parties.

C. Privilege Assignment

This protocol is run by the patient's PC to assign the secret keys generated in *private PHI storage* to the family and P-device, so that they can search over the stored PHI. We need this privilege assignment as the backup to successfully retrieving the patient's PHI during emergencies. Specifically, the patient has the freedom to grant privileges to his family and P-device to search over his PHI when he is unable to do so. The patient can correspondingly revoke these entities to suspend the searching privilege (e.g., when P-device is lost). Furthermore, the patient should be the sole party to update or modify his stored PHI while the privileged entities should merely be able to perform searches. These requirements can be easily fulfilled by adding two algorithms ASSIGN, REVOKE to the SSE primitive, according to [17]. ASSIGN takes place between the patient and privileged entities U:

$$patient \rightarrow U : E'_\mu(TP_p \parallel \nu \parallel a \parallel b \parallel c \parallel d \parallel SI \parallel KI \parallel dictionary \parallel s \parallel X), t_2, HMAC_\mu(E'_\mu \parallel t_2),$$

where μ is the pre-shared secret key between *patient* and U (one key for each entity in U), X denotes the secrets needed for BE. We shall point out that entities in U can be assigned the same TP_p/Γ_p or a different pair generated by the patient as mentioned above. A different pair for each entity has the merit of preventing S-server and other malicious parties from linking all activities under the same TP_p. REVOKE is executed between the patient and S-server to update d:

$$patient \rightarrow S\text{–}server : E'_\nu(d' \parallel BE'_{U'}(d')), t_3, HMAC_\nu(E'_\nu \parallel t_3),$$

where $d' \in_R \{0,1\}^k$, and U' is the new set excluding the revoked entities. *S-server* check the integrity of the message and replaces d, $BE_U(d)$ with d', $BE'_{U'}(d')$. Note that in *private PHI storage*, the interactions between *patient* and U (i.e., sending θ), and between *patient* and *S-server* (i.e., sending θ, d, $BE_U(d)$) take the same secure procedures as described above. We omitted the details there due to space limitation.

D. Common-Case PHI Retrieval

On subsequent visits to the hospital, the patient will be asked for his PHI relevant to the treatment he seeks. In previous sections, we mentioned that the patient stored necessary information for future PHI retrieval, and designed protocol for private PHI storage, both of which are critical for the protocol in this section (and next) since they enable the patient to retrieve the portion of PHI pertinent to his treatment instantly upon the physician's request. Storing PHI in online central servers facilitates the patient, or more importantly, the P-device as introduced next during emergencies, to perform timely searches and retrieval, possibly across hospitals.

The protocol is executed between the patient and S-server as follows:

1. $patient \rightarrow S\text{–}server : TP_p, SI, TD(kw), t_4, HMAC_\nu(TP_p \parallel SI \parallel TD(kw) \parallel t_4)$,
2. $S\text{–}server \rightarrow patient : \Lambda(kw), t_5, HMAC_\nu(\Lambda(kw) \parallel t_5)$,

where $TD(kw) = (\Re_c(kw), f_b(kw))$ is the trapdoor for keyword kw. After step 1, S-server performs an algorithm SEARCH locally by computing $\delta = T[\Re_c(kw)]$, $\upsilon = \delta \oplus f_b(kw) = (addr \parallel \lambda)$, where $addr$ and λ are used to locate and decrypt the linked list for kw. The algorithm finally outputs $\Lambda(kw)$, the set of encrypted files containing kw. The output is returned to the patient as in step 2. Note that multiple keywords can be searched in step 1 with corresponding files returned in step 2. The patient then decrypts $\Lambda(kw)$ using $E'^{-1}_s(\Lambda(kw))$ on his cell phone and sends the plaintext PHI to the physician.

The key point of adopting the keyword search is the small number of files (instead of the entire file collection) returned to the patient, which fits the EHR system elegantly according to the privacy requirement for disclosing only minimum necessary health information. The cell phone would suffice due to the low complexity of the retrieval protocol. Note that we assume the patient also incorporates into the keyword index the network address information of S-servers for each stored PHI file collection. The protocol execution remains the same for retrieval across hospitals, except for the shared key which is derived in the HIBC domain.

E. Emergency Health information retrieval

This protocol is designed to handle the emergency case in which the patient is physically incompetent to perform PHI retrieval for treatment. We propose two approaches, the first of which leverages *family*.

1) Family Based Approach: It is intuitive and common practice to seek help from a family member that serves as the emergency contact and will most likely be available when the patient encounters emergency. As shown in *privilege assignment*, *family* is equipped with all necessary information to execute *common-case PHI retrieval* in an analogous procedure to that executed by *patient* with one more round of interactions.

1. $family \rightarrow S\text{–}server$: TP_p, m, t_6, $HMAC_\nu(TP_p \parallel m \parallel t_6)$,
2. $S\text{–}server \rightarrow family : BE_{U'}(d)$, t_7, $HMAC_\nu(BE_{U'}(d) \parallel t_7)$,
3. $family \rightarrow S\text{–}server : SI$, $TD_U(kw)$, t_8, $HMAC_\nu(SI \parallel TD_U(kw) \parallel t_8)$,
4. $S\text{–}server \rightarrow family : E'_s(kw)$, t_9, $HMAC_\nu(E'_s(kw) \parallel t_7)$,

where m is the request to obtain $BE_{U'}(d)$ from which the most up-to-date d can be recovered only by non-revoked entities in U' using X, $TD_U(kw) = \theta_d(TD(kw))$ is the trapdoor computed by *family*. Before applying SEARCH to return $\Lambda(kw)$, *S-server* recovers $TD(kw) = \theta_d^{-1}(TD_U(kw))$ and checks its validity.

The essence of family based approach captures the security factor of "someone you trust" [27], the key advantage of which is that "someone" has subjective judgments to avoid possible security breaches. In our context, the family is able to judge if the physician requesting the patient's PHI has appropriate access rights, and if a particular physician has leaked the PHI illegally, without exercising any security mechanisms. It greatly reduces the complexity of our system design. However, the drawback of this approach is also significant, that is, the availability and timeliness of the family's presence at any emergencies cannot be

guaranteed, which could be fatal in our scenario. As a result, we propose a second line of defense leveraging P-device should the family based approach fail.

2) P-Device Base Approach: First and foremost, we clarify that our system is by no means able to deal with all possible emergencies. Instead, our protocols are devoted to cover patients that are most likely or highly risky to experience sudden emergencies, by which we mean the emergencies that are unforeseeable. For instance, patients suffering from heart attacks or failures are highly susceptible since future attacks/failures cannot be predicted. On the other hand, patients in the their last few weeks of pregnancies are much easier to care for since the family will be well prepared during that period of time, in which case the family based approach would be most effective. As healthcare technology evolves, patients with high risk diseases are obtaining medical aids from more advanced equipments, such as sensors and PDAs in body sensor networks for monitoring critical health issues, the IMD (implantable medical device) implanted in bodies to assist in proper functioning of organs, etc. It is therefore reasonable to assume the presence of such equipments, the so-called *P-device* in HCPP, carried or worn by patients who are to encounter sudden emergencies. Note, however, that we can extend the notion of P-device in our system to incorporate smartphones, or any portable devices with required capabilities, so that patients without monitoring equipments can also be covered by the P-device based approach. We do not pursue further on this issue but argue that a vast majority of emergencies can be properly handled by our two approaches.

The physician may be interested in two types of health information upon arriving at the emergency scene: PHI and MHI. P-device should be programmed with an emergency functionality, or simply has an emergency button. The physician pushes the button and P-device enters the emergency mode, in which P-device automatically connects to A-server through wireless network access. Meanwhile, the physician contacts A-server to authenticate as the emergency caregiver on duty. This can be achieved, for example, by having the physician sign in at his hospital for work and sign out when he leaves, so that the list of "today's on-duty physicians" of each hospital can be published online for A-server to check against.

PHI Retrieval: For the physician with identity ID_i to obtain the patient's PHI:

1. *physician* $\rightarrow$ *A–server*: ID_i, m', t_{10}, $IBS_{\Gamma_i}(ID_i \parallel m' \parallel t_{10})$,
2. *A–server* $\rightarrow$ *physician*: $E'_{\varpi}(nounce)$, t_{11}, $IBS_{\Gamma_{A-server}}(ID_i \parallel TP_p \parallel E'_{\varpi} \parallel t_{11})$,
3. *A–server* $\rightarrow$ *P–device*: ID_i, $IBE_{TP_p}(ID_i \parallel nounce \parallel t_{11}), t_{11}, IBS_{\Gamma_i}(ID_i \parallel TP_p \parallel IBE_{TP_p} \parallel t_{11})$,

where m' is a request for a one-time passcode, ϖ is the shared key between *physician* and *A-server* which can be derived locally by both parties as $\varpi = e(\Gamma_i, PK_{A-server}) = e(PK_i, \Gamma_{A-server})$ with $PK_x = H_1(ID_x)$, $nounce \in_R Z_q^*$ is a random secret generated by A-server, IBS is the identity-based signature [28]. The last two steps take place simultaneously and only after the physician successfully authenticates himself as the emergency caregiver on duty. P-device then prompts the physician to enter his ID, and $nounce$ as the one-time passcode. If the passcode is valid, P-device prompts the physician to enter the keywords $KW = \{kw_1, ..., kw_n\}$ he needs to search. If the keywords result in a match in the dictionary, P-device proceeds to execute the PHI retrieval with S-server as in the family based approach. Finally, the A-server generates a trace $TR = (ID_i, TP_p, t_{10}, t_{11}, IBS_{\Gamma_i})$ to record this transaction. P-device generates a record $RD = (ID_i, TP_p, KW, t_{11}, IBS_{\Gamma_{A-server}})$ as the evidence that ID_i and TP_p had a transaction at time t_{11} with A-server's signature. KW is included for the patient to later decide if the physician performed only necessary and relevant searches.

MHI Retrieval: Where applicable, the physician would also like to obtain the MHI which records the patient's recent monitored health data and would most possibly imply the cause of the sudden emergency (e.g., irregular heartbeat intervals, sudden surge in blood pressure). Our HCPP adopts the role-based technique [7] enabled by IBE. The MHI is collected and encrypted by P-device using an identity string ID_r, for which only the A-server can generate the corresponding private key Γ_r at a later time. Since the emergency caregiver is unpredictable, ID_r can be a general descriptive string like "$Date \parallel Duty \parallel ServiceArea$". The MHI collected on a particular day can be made searchable for each of the following, say, 5 days within which the MHI will be needed should emergency occur. P-device stores the encrypted MHI in S-server in advance for future searches as:

$$P\text{–}device \rightarrow S\text{–}server: TP_p,$$
$$BE_{ID_r}(MHI) \parallel PEKS_{\sigma}(ID_r, kw), t_{12},$$
$$HMAC_{\nu}(TP_p \parallel IBE_{ID_r} \parallel PEKS_{\sigma} \parallel t_{12}),$$

where $PEKS_{\sigma}(ID_r, kw) = (\sigma P, H_3(e(H_2(kw), PK_r)^{\sigma}))$ is the searchable public key encryption with $\sigma \in_R Z_q^*$ chosen by *P-device*. The keyword kw can be the period of time (i.e., the 5 days) for retrieving the most relevant health data. The choice of keywords (also in the PHI retrieval) must obey an agreed-upon syntax so that the physician will be able to specify proper keywords for searching. The single keyword PEKS shown above can be easily extended to enable multiple-keyword search [29]. After authenticating himself and obtaining Γ_r from A-server following the above steps 1, 2, 3, the physician and S-server interact as follows:

1. *physician* $\rightarrow$ *S–server*: ID_r, $TD_r(kw)$, t_{13}, $HMAC_{\rho}(ID_r \parallel TD_r \parallel t_{13})$,
2. *S–server* $\rightarrow$ *physician*: $IBE_{ID_r}(MHI)$, t_{14}, $HMAC_{\rho}(IBE_{ID_r} \parallel t_{14})$,

where $TD_r(kw) = \Gamma_r \cdot H_2(kw)$ is the trapdoor computed by *physician* for searching kw, and $\rho = e(\Gamma_r, PK_{S-server}) = e(PK_r, \Gamma_{S-server})$ is the shared key derived locally by *physician* and *S-server*. *S-server* searches over the encrypted MHI designated for *physician*, and returns the results $IBE_{ID_r}(MHI)$ to *physician*. We have assumed for simplicity that the S-server also stores MHI in addition to PHI. In practice, nonetheless, there exists a monitoring center possibly different from S-servers for storing the monitored data.

V. Security and Efficiency Analysis

A. Security Analysis

In this section, we show that the proposed HCPP system satisfies the security requirements set in Section III.

Privacy and Confidentiality: First of all, all PHI (and MHI) files are encrypted which prevents the storage server and other malicious parties to learn the content of the PHI, achieving

both patient privacy and PHI data confidentiality. Second, the unlinkability property of the privacy requirement is guaranteed by having the patient, family, or P-device actively control the retrieval of the encrypted PHI from S-server and return plaintext PHI to the physician, based on the SSE and PEKS primitives, breaking the link present in [11] where the physician can directly query the server. Moreover, the distributions of the secret keys in *privilege assignment* and the nounce in *emergency health information retrieval* are through secure encryption schemes to provide confidentiality for sensitive messages.

Fail-open: We developed family based and P-device based approaches as the backup mechanisms for emergency situations. Both approaches are effective in successfully retrieving the needed PHI in the absence of the patient and preserve the privacy properties as described above.

Access Control: The fact that in our HCPP system the physician must always request the patient, family, or P-device for accessing the PHI facilitates access control. The patient and family can reject inappropriate access requests by subjective judging. P-device relies on A-server as a trusted authority to verify the eligibility of the physician for accessing both PHI and MHI. As a result, only physicians with certain access rights can learn the content of PHI or MHI through legitimate requests.

Accountability: This requirement can be easily satisfied when either patient or family is executing the protocols due to the assumption we made earlier that possible sources of PHI leakage can be recalled. When the P-device based approach is leveraged, the patient can check back his P-device after the emergency is resolved to obtain the records (RDs) created in *emergency health information retrieval*. RDs contain information on which physicians interacted with P-device at what times, necessary for the patient to contact A-server (with A-server's signature as proof), to obtain the traces (TRs, with physician's signature) from A-server as evidence to hold the physician(s) accountable. Even if the PHI is not leaked, the patient can check the keywords in the RDs to determine if the physician should be held accountable for searching any PHI other than appropriate.

Data Integrity: Since the PHI and MHI are encrypted, no one except the patient himself can perform any meaningful modifications. Although the actual modifications to the PHI are performed by an authorized physician, the patient must agree and retrieve the plaintext PHI for the physician. Note that it is technically possible for the family and P-device to retrieve the plaintext PHI for a physician to modify. However, the family or P-device would not be able to store the modified PHI back, which involves a file collection update and can only be performed with the patient's secret key (not shown in the paper) not distributed to *any other* entity. The protocol message integrity is ensured by including HMAC or digital signatures in the message exchanges.

Availability: When the patient or family is available, the S-server that stores the desired PHI can be looked up using the keyword index. In the protocol description, we implicitly assumed that the S-server is inside its parent A-server's domain (cf. Section IV. A) so that the standard IBC suffices. As mentioned in *system setup*, the hierarchical IBC (HIBC) will be used if the S-server is outside its parent A-server's domain. The patient can be provided a temporary key pair (similar to TP_p/Γ_p) at level 3 of the hierarchical tree, enabling the patient to interact with any S-server throughout the country. In emergencies, the interactions between the physician or P-device and any A-server can be carried out similarly. The HIBC infrastructure ensures the availability of desired PHI wherever the PHI is stored.

B. Efficiency Analysis

1) Storage: The major storage cost is due to patients' PHI storage, for which our SSE-based *private PHI storage* protocol is developed to achieve efficiency at both the patient and server side. Since the PHI files are outsourced to S-server, the patient has $O(1)$ (i.e., constant) storage in terms of the retrieval-related information (i.e., his entire PHI file collection is encrypted and stored using the same set of secrets). The MHI storage puts no extra overhead on the patient side except a dictionary of keywords for PEKS. In addition, the patient needs to store the key pair TP_p/Γ_p (2 $|G_1|$ elements) and several shared keys ($|G_2|$ elements) for protocol interactions, which are in total several hundred bytes and can be handled easily even by low-end mobile devices. The storage requirement on the S-server is $O(N)$ (i.e., linear) with N the number of PHI files in a collection, for each patient, which is the best achievable so far by searchable symmetric storage schemes preserving privacy (cf. Table 1 in [17]). As explained before, MHI is usually stored in a designated monitoring center, for which the storage costs should not be a concern.

2) Communication: The *private PHI storage* protocol introduces no communication overhead except the one-time transmission of encrypted PHI files and associated secure index to S-server. The *privilege assignment* protocol also involves only one transmission to S-server, with the rest communications (i.e., distribution of the retrieval-related information) taking place locally. Most communication overhead is incurred in the health information (PHI and MHI) retrieval (cf. Section IV. D, E), which, however, is expected to be infrequent since visiting hospitals for treatment is not a daily activity. During the common-case PHI retrieval, the communication is one round and the exchanged data are of small size (e.g., the returned PHI files containing a particular keyword are of small size relative to the entire file collection). The emergency health information retrieval executed by the family and P-device are more communication intense due to the extra steps of obtaining the secret key from S-server, and the role-based authentication leveraging A-server (P-device based approach only), which is the inevitable consequence of providing security guarantees. Indeed, the extra round of communication with S-server to recover the secret key can be omitted (i.e., remove the functions ASSIGN and REVOKE in *privilege assignment*), rendering the revocation of the lost P-device for privacy-preserving purposes (cf. Section VI. A) intractable. Similarly, if the role-based authentication is eliminated, P-device could merely authenticate with the physician (on duty or not) without being able to exercise access control. Considering the fact that our protocols introduced only one more round of communication for each of the above security add-ons, the resulting design is very reasonable and strikes a good balance between security and efficiency.

3) Computation: Computation and storage efficiency are intrinsically related at the S-server side, in that a well designed storage algorithm gives rise to efficient searches for hit. The design of the lookup table T in the secure index exploits the algorithm in [30] and enables S-server to return the desired PHI files in $O(1)$ time. At the patient side, computation mainly occurs during private PHI storage and subsequent protocol message exchanges

in which only computationally-efficient symmetric key operations (encryption, decryption, HMAC) need to be performed. The only public key operation performed by the patient and P-device is the IBE and PEKS for MHI storage, respectively, where expensive pairing computations are involved. Fortunately, IBE and PEKS encrypted MHI files are for future emergency uses and can be pre-computed (offline). P-device, however, needs to carry out two public key operations including an IBE decryption and an IBS verification in the role-based authentication, which, unlike PEKS, cannot be performed offline. With pre-computation, P-device computes two pairings for both operations. Using the results in [31], the time taken for computing a Tate pairing is around 20ms for a similar level of security to 1024-bit RSA. The timing has been significantly improved recently [32], rendering a wider application of the Tate pairing technique and a more acceptable computation complexity at P-device.

VI. Attacks and Countermeasures

A. Collusion

In our HCPP system, the goal of collusion is to learn the content of a chosen patient's PHI (i.e., compromise this patient's privacy). We only consider possible collusion among entities that can fulfill this goal. The entities involved in HCPP protocols are *patient*, *family*, *P-device*, *physician*, *S-server*, and *A-server*. Obviously, *patient* will not collude to attack himself, and it is extremely unlikely for *family* to launch such attacks. Therefore, collusion attacks will be launched among the remaining four entities.

Although *P-device* belongs to the patient and would normally not engage in attacks, an outsider can compromise P-device in case it is lost or stolen. The patient can simply revoke P-device as described in *privilege assignment* once he realizes the loss, before which the outsider will have chance to attack. An interesting observation is that the outsider in fact has much higher success rate to compromise the patient's privacy alone than engaged in collusion. For an unsophisticated outsider to learn the PHI using *P-device*, he must find a corrupt emergency caregiver on duty to input an identity and passcode for *P-device* to enter the emergency retrieval mode. Alternatively, he must find *any* corrupt physician and a corrupt A-server to fake the role-based authentication. Most importantly, the outsider must find such colluding candidates before the patient revokes *P-device*. A sophisticated outsider, on the other hand, can re-program *P-device* to skip the input phase and collude with any physician to enter valid keywords for searches. However, the sophisticated outsider will not opt for such collusion when he can simply compromise P-device and obtain all the stored information necessary for performing the PHI retrieval. This last option is least time-consuming for a sophisticated attacker and is apparently of highest success rate before the patient can revoke *P-device*. Thwarting this attack is a non-trivial task which is an open problem in any system where revocation is involved, since the vulnerable period before the credential can be revoked creates opportunities for attacks. One common approach is to employ the tamper proof module (TPM) on *P-device* which erases all secrets upon detecting tampers. In our scenario, we can program *P-device* to send message alerts to the patient's cell phone or email address whenever the PHI-retrieval related secrets are accessed so that the patient will notice the loss of his P-device if it is not a true emergency. *P-device* can also send the records RDs whenever they are created in case the lost *P-device* cannot be regained. The privacy-preserving location tracking technique for lost or stolen devices [33] is a viable solution for regaining the lost device, which however requires extra mechanism.

S-server is a "useless" entity to collude with in that it is merely used for searching and returning encrypted files. If an entity B possesses all necessary information for searching (like *patient*, *family*, *P-device*), B can successfully retrieve the desired PHI without colluding with *S-server*. In contrast, if B does not possess all such information (like *physician*, *A-server*), colluding with *S-server* will not gain B any advantage in retrieving the PHI. Perhaps the only reason to collude with *S-server* is to exploit previous searches performed by the same patient through traffic analysis, the countermeasure of which will be discussed shortly.

The collusion of *physician* and *A-server* (with or without *S-server*) cannot result in the retrieval of desired PHI since neither entity has the right information. The only chance is as mentioned above to collude with an unsophisticated outsider (i.e., who is incapable of altering or tampering with the internal of P-device) to exploit the secrets stored in *P-device*. This collusion attack can be defeated using the RDs recorded in P-device, which contain the physician's identity and A-server's signature, together with "today's on-duty physicians" list accessible online, rendering the colluding parties not exculpable of their launched attack.

B. Traffic Analysis

Traffic analysis attack falls into the following categories:

1. Attackers exploit previous searches performed by the patient over the encrypted PHI to profile the search pattern of each patient.
2. Attackers trace the network address of the patient's PC or cell phone to identify the owner of the stored PHI files.

In Category 1, previous searches can leak a) the memory addresses of previously returned files containing a particular keyword, and b) whether two searches were for a same keyword. The information in a) bears certain ambiguity in that a) can result from searches under different keywords, and thus the attackers cannot learn which keyword was searched for or perform profiling on the search pattern. Regarding b), there are well established schemes [15], [16] to hide this information with lower efficiency. In our context, the patient can make the keyword choice flexible such that multiple keywords can be used in different searches leading to the same set of files, with the added complication in the size increase of the keyword index, and the encryption and storage of more PHI files.

Category 2 can be coped with by building our HCPP system on an anonymous underlying network such as Tor [34], [35] which provides anonymous communication channels between the patient and S-server. The obfuscation of patients' traffic effectively prevents the attackers from tracing the traffic origin and identifying the owner of the PHI data. Furthermore, the patient can self-generate multiple key pairs TP_p/Γ_p for different searches, so that his successive activities will not be linked under the same TP_p.

C. Timing Analysis

Timing analysis is performed by powerful attackers who can follow the routine of the patient, narrowing down the time range when the patient will upload his PHI files (e.g., after the patient returns from the hospital to create or update his PHI data).

The most effective countermeasure may be to employ some scheduling technique to randomize the uploads and minimize the correlation. A PRF or PRG (pseudo-random generator) with a random seed would suffice for the task.

D. Denial of Service (DoS)

DoS attack is not strange to any system with centralized points. Though not specific to HCPP, DoS attack can disturb the system operation by restraining S-servers from receiving PHI uploads, or disabling A-servers to perform timely authentication. S-servers are distributed across the area and the attackers must be powerful enough to disable a large number of S-servers. Despite the assumption that S-servers will not arbitrarily delete the stored files, they can do so when detecting abnormalies since an honest patient's PHI data are usually trivial in comparison to the storage capacity of S-servers. The A-servers, however, are much more centralized and susceptible to DoS attacks. The attack to A-servers can be addressed by splitting the role of an A-server to several local offices, and utilizing the hierarchical IBC architecture in HCPP for convenient cross-domain authentication (e.g., the physician can call the toll-free number to access another A-server if the one in his domain is unreachable).

VII. Conclusion

In this paper, we design a secure EHR system to protect patient privacy and enable emergency healthcare. The system is demonstrated to be resilient to various attacks, fulfill the desired functionalities, satisfy the security requirements, and maintain a good balance between security and efficiency.

Acknowledgment

This work was partially supported by the U.S. National Science Foundation under grants CNS-0916391 and CNS-0716450 and the National Natural Science Foundation of China under grant No. 61003300. The work of X. Zhu was also partially supported by the Fundamental Research Funds for the Central Universities under grant No. JY10000901021 and China 111 project under grant B08038 with Xidian University, Xi'an, China.

References

[1] G. M. Stevens, "A brief summary of the medical privacy rule," *CRS Report for Congress*, 2003.

[2] "Electronic health record," *http://en.wikipedia.org/wiki/Electronic_health_record, retrieved 8/18/2008*.

[3] M. C. Rash, "Privacy concerns hinder electronic medical records," *The Business Journal of the Greater Triad Area*, Apr. 2005.

[4] R. Pear, "Warnings over privacy of u.s. health network," *New York Times*, Feb. 2007.

[5] P. Ray and J. Wimalasiri, "The need for technical solutions for maintaining the privacy of EHR," *in Proc. 28th IEEE EMBS Annual International Conference*, pp. 4686–4689, Sept. 2006.

[6] U. Sax, I. Kohane, and K. D. Mandl, "Wireless technology infrastructures for authentication of patients: PKI that rings," *Journal of the American Medical Informatics Association*, vol. 12, no. 3, pp. 263–268, 2005.

[7] M. C. Mont, P. Bramhall, and K. Harrison, "A flexible role-based secure messaging service: Exploiting IBE technology for privacy in health care," *in Proc. 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, 2003.

[8] T. Denning, K. Fu, and T. Kohno, "Absence makes the heart grow fonder: New directions for implantable medical device security," *in 3rd USENIX Workshop on Hot Topics in Security (HotSec'08)*, July 2008.

[9] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," *in Proc. IEEE Symposium on Security and Privacy*, May 2008.

[10] W.-B. Lee and C.-D. Lee, "A cryptographic key management solution for HIPAA privacy/security regulations," *IEEE Trans. Information Technology in Biomedicine*, Jan. 2008.

[11] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: an identity-based cryptography approach," *The ACM Conference on Wireless Network Security (WiSec'08)*, Apr. 2008.

[12] L. Zhang, G. J. Ahn, and B. T. Chu, "A rule-based framework for role-based delegation and revocation," *ACM Transactions on Information and System Security*, vol. 6, no. 3, pp. 404–441, 2003.

[13] S.-D. Bao, Y.-T. Zhang, and L.-F. Shen, "Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems," *in Proc. 28th IEEE EMBS Annual International Conference*, pp. 58–65, Sept. 2005.

[14] D. Boneh and M. Franklin, *Identity-based encryption from the weil pairings*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp. 514-532, Springer-Verlag, 2001.

[15] R. Ostrovsky, "Software protectoin and simulations on oblivious RAMs. MIT Ph.D. Thesis, 1992," *in Proc. 22nd Annual ACM Symposium on Theory of Computing*, 1990.

[16] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious RAMs," *Journal of the ACM*, pp. 431–473, 1996.

[17] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *in ACM Conference on Computer and Communications Security (CCS)*, 2006.

[18] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *in EUROCRYPT 2004, LNCS 3027. Springer*, 2004.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing. extended abstract in CRYPTO 2001," *SIAM J. of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[20] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions," *in V. Shoup, editor, Crypto 2005, volume 3621 of LNCS. Springer*, 2005.

[21] C.-H. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu, "GAnGS: Gather, authenticate 'n group securely," *Proc. ACM Annual Int'l Conf. on Mobile Computing and Networking (MobiCom)*, Sept. 2008.

[22] C.-H. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu, "Seeing-is-believing: Using camera phones for human-verifiable authentication," *in Proc. IEEE Symposium on Security and Privacy*, May 2005.

[23] L. Zhong, M. Sinclair, and R. Bittner, "A phone-centered body sensor network platform: cost, energy efficiency & user interface," *in Proc. International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, 2006.

[24] "45 C.F.R. part 160 - general administrative requirements, 160.103: definitions," *Department of Health and Human Services*, vol. 1, Oct. 2002.

[25] J. Sun, C. Zhang, and Y. Fang, "A security architecture achieving anonymity and traceability in wireless mesh networks," *IEEE Conf. on Computer Communications (INFOCOM)*, pp. 1687–1695, Apr. 2008.

[26] A. Menezes, P. V. Oorschot, and S. Vanston, *Handbook of Applied Cryptography*, Boca Raton, CRC Press, 1996.

[27] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: somebody you know," *in ACM Conference on Computer and Communications Security (CCS)*, 2006.

[28] F. Hess, *Efficient identity-based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

[29] J. Baek, R. Safiavi-Naini, and W. Susilo, *Public key encryption with keyword search revisited*, Cryptology ePrint Archive, Report 2005/191, available at http://eprint.iacr.org/2005/191.pdf, 2005.

[30] M. L. Fredman, J. Komlós, and E. Szemerédi, "Storing a sparse table with O(1) worst case access time," *Journal of the ACM*, vol. 31, no. 3, pp. 538–544, 1984.

[31] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *CRYPTO 2002, Springer-Verlag, LNCS 2442*, pp. 354–368, 2002.

[32] P. S. L. M. Barreto, S. D. Galbraith, C. ÓhÉigeartaigh, and M. Scott, *Efficient pairing computation on supersingular abelian varieties*, Cryptology ePrint Archive, Report 2004/375, available at http://eprint.iacr.org/2004/375.pdf, Sept. 2005.

[33] T. Ristenpart, G. Maganis, A. Krishnamurthy, and T. Kohno, "Privacy-preserving location tracking of lost or stolen devices: cryptographic techniques and replacing trusted third parties with DHTs," *in 17th USENIX Security Symposium*, pp. 275–290, July 2008.

[34] R. Dingledine, "Tor: An anonymous internet communication system," *Workshop on Vanishing Anonymity, The 15th Conf. on Computers, Freedom, and Privacy*, Apr. 2005.

[35] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," *in Proc. USENIX Security Symposium*, pp. 303–320, Aug. 2004.