CHAPTER 27

Security and Privacy for Mobile Health-Care (m-Health) Systems

Jinyuan Sun, Xiaoyan Zhu, Chi Zhang, Yuguang Fang

27.1. INTRODUCTION

Fast and secure access to patients' records helps to save lives with timely treatment in emergency situations. Therefore, anywhere-anytime-accessible online health-care or medical systems play a vital role in daily life. Advances in (wireless) communications and computing technologies have lent great forces to migrating health-care systems from the paper based to the EHR (electronic health record) based, giving rise to increased efficiency in human operations, reduced storage costs and medical errors, improved data availability and sharing, etc. Unfortunately, such convenience also comes with concerns, which should be carefully addressed. For example, medical or health record privacy is a major concern to the patients and becomes the major barrier in the deployment of the EHR-based health-care systems. It is observed that privacy and security breaches have already penetrated every aspect of our activities and living environment including health care, financial, voting, e-commerce, military, etc. Thus, there is an urgent need for the development of architectures assuring privacy and security that are imperative to safeguarding confidential information wherever it digitally resides. Despite the paramount importance, little progress has been introduced by researchers in the design of security and privacy architectures for the EHR-based health-care system. In particular, two extremely critical issues are rarely touched in the research realm: health information privacy and sharing.

Health information privacy (or medical record privacy) refers to the confidentiality and access restrictions of patients' protected health information (PHI) which contains sensitive and 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 aware of the disease history of a patient, a person with certain types of disease may be discriminated by the health-care provider, and so on. However, fundamental developments of health-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 records. Computers (connected to a network) are now commonly used by the health-care providers to store and retrieve patients' electronic health records (EHRs).

Handbook on Securing Cyber-Physical Critical Infrastructure. DOI: 10.1016/B978-0-12-415815-3.00027-3

EHR systems are used in lieu of paper systems to increase physician efficiency, to reduce costs (e.g., storage) and medical errors, to improve data availability and sharing, etc. An exemplary successful implementation of the EHR system in the United States is the Veterans Administration health-care system, with over 155 hospitals and 800 clinics. It is one of the largest integrated health-care 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 health-care systems. Statistical results of the actual adoption rate of EHR in U.S. medical systems can be found in Ref. [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. This was reflected in a nationwide survey conducted in February 2005 by Harris Interactive of Rochester, NY, in which 70% of the population were concerned that personal medical information would be leaked because of weak data security [3]. This sentiment has undoubtedly been exacerbated by illegal disclosures including: Christus St. Joseph Hospital, Houston, Texas (16,000 records were compromised by theft), University of Chicago Hospital (employees were found selling patient data), and Wilcox Memorial Hospital, Kauai, Hawaii (130,000 records were compromised by theft) [4]. Records stored in a central server and exchanged over the Internet are subject to theft and security breaches. The Health Insurance Portability and Accountability Act (HIPAA) in the United States was established to regulate EHR-related operations. Privacy issues are particularly not addressed adequately at the technical level [4]. Therefore, in addition to governmental regulations, standardization and an overall strategy are needed to ensure that privacy protection would be built into computer networks linking insurers, doctors, hospitals, and other health-care providers [5]. The implementation of the standardization or strategy will undoubtedly be relying on technical details, which are rarely studied in the research community, leaving us numerous research opportunities.

Health information sharing takes place as daily routine where the primary caregiver (e.g., family doctor) frequently refers patients to specialists who are unavailable at the primary caregiver's organization. The sharing also occurs as a result of cross-organizational (or simply, cross-domain) collaborative research for studying diseases and improving clinical care or collaborative remote surgical operations. Central issues around the sharing of health information include authentication of an entity, delegation of access rights and permissions, access control to patient records, and revocation of access rights and permissions with respect to an outside collaborator. In its original form, delegation of rights is used to appoint a proxy signer who signs on behalf of the delegator in case he/she is absent. In our EHR system, delegation of rights can be used to allow the delegatee's access to shared health records. More challenging yet, such access should also be restricted to only the portion(s) of data intended for sharing, since illegal disclosure of highly confidential data such as a patient's health records can be devastating and is against the HIPAA regulations. Moreover, revocation must be dynamic in that the delegator should be able to revoke delegated rights at any time due to unforeseeable situations. Data sharing in a distributed fashion, e.g., shifting the delegator's task to each delegatee and thereby making the delegation process transparent to the delegator, allowing each cooperating health-care provider to process data locally for either treatment or research, delivers tremendous benefits including higher efficiency, better scalability, and lower complexity at the user end due to reduced human intervention, etc. Designing a secure EHR system that offers information sharing capability and guarantees health record privacy is equivalent to gluing all the aforementioned challenges together and providing a feasible solution, which is by no means a trivial task and will be the focus of this chapter.

The majority of works on privacy protection in health-care systems still concentrate on the framework design or solution proposals with little or no technical realization [4, 6–12]. 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 health-care systems becomes clearer and more stringent, a few recent works followed this line of research. Lee and Lee [13] proposed a cryptographic key management solution for privacy and security regulations regarding patients' PHI. Although technically correct, the proposed scheme is unreasonable because 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. [14] is a technical realization of the role-based approach proposed in Ref. [7]. In spite of specifying the algorithms for storing and retrieving health-care 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 a querying doctor. Such leakage will compromise patients' privacy by violating the unlinkability requirement.

A survey on delegation for information sharing in distributed health-care context is given in Ref. [15] with no specific technical design to cope with the major issues identified in this context, namely, least-privilege delegation, revocation, onward delegation, and dynamically changing credentials. Current approaches to delegation in distributed health-care context are identified and categorized as proxy certificates [16–19], call-backs [20], XML [21–24], and role-based delegation [10–12]. Among these approaches, proxy certificates and role-based delegation coincide in ideas with some part(s) of our proposed solution. However, proxy certificates bear some key drawbacks, i.e., static access control and revocation, that render this approach not readily applicable. Role-based delegation is free of these drawbacks but has problem in ensuring least-privilege assignment, due to the lack of fine-grained access control. Implementing the proposed information sharing procedure using standard XML language and framework is parallel to our work and can be considered for future work (e.g., XML framework can be used as a standard means to specify the instantiation of the signatures used in the technical descriptions of our system. Finally, there are a few projects on system or architectural design in the health-care field, including cryptographic and system aspects of medical information privacy assurance [25], self-scaling and distributed health information architecture [26], and secure grid-enabled health care [27], which focus on vastly different aspects of health-care system and are largely parallel to our work.

In this chapter, we present our design of a security architecture, HIPS, for EHR systems based on cryptographic tools. First, the proposed system should offer both full privacy for patients without escrow (e.g., the trusted server in Ref. [13]), and the capability of handling emergency situations, which are intrinsically related and somehow contradictory. Full privacy means even when the patient is incapable of authorizing access to his/her PHI during emergency, 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 health-care system and must be carefully coped with. Second, information sharing capabilities should be provided by leveraging authentication, delegation, access control, and revocation. Authentication is a fundamental requirement prior to privacy guarantee and health information sharing, to assure the authenticity of a communicating identity. Delegation leverages proxy signature and role-based approach to distribute the task among delegatees (more specifically, delegation servers of delegatees' organizations) and ensure transparency for the delegator. Access control is an enhancement to delegation for further and accurately restricting the data portions accessible to the delegatee. The revocation mechanism describes a dynamic approach to tackle unexpected or urgent revocation issues. The ultimate goal of the architecture described in this chapter is to simulate, implement, and test HIPS to obtain the first functional, secure, and private EHR-based health-care system, with the following design objectives:

1. Establishing trust: leveraging hierarchical identity (ID)-based cryptography (HIBC) for authentication and key management.
2. Protecting patient privacy: featuring patient-controlled health information for betterprivacy and compliance with HIPAA regulations.
3. Controlling access to patients' health records: bearing different design requirements for the access to different portions of patients' health data, with those containing identification information subject to finer control.
4. Sharing information for health care and research: exploiting proxy signature and role-based delegation for secure cross-domain collaborations.
5. Revoking delegated rights: providing viable means for terminating ongoing collaborations once violation of rules is detected.
6. Resolving conflicts of security and functional requirements: enabling life-saving treatment during emergency while not compromising the privacy of patients' health data.

In a nutshell, the proposed security architecture for m-health has significance in many aspects. Securing cyberspace is one of the top priorities in protecting our national infrastructure including the health-care systems. Protecting citizens' privacy is critical to the deployment of such application systems where highly sensitive and confidential information is involved. EHR systems have been envisioned to be the most viable solutions to deliver efficient health care, simplified management, and seamless information sharing and will be the next big business push. The current EHR systems have not addressed well on privacy and information sharing, and the proposed solution may lay the foundation for online fast access to patients' record in emergency situations or collaborative diagnosis, hence can potentially save people's lives with proper and timely treatment.

27.2. ELECTRONIC HEALTH RECORD (EHR)

Electronic health record refers to a patient's medical record created, stored, transferred, and accessed digitally, as opposed to the traditional paper-based health record. EHR is the central piece of information in realizing electronic health care and may record medical data such as radiology images (CAT, MRI, and X-ray), laboratory test results, medication, allergy, disease history, billing information, as well as some processed or aggregated medical data (ECG, emergencies, critical health conditions, etc) monitored by wireless body sensor networks.

EHR systems are used in lieu of paper systems to increase physician efficiency, reduce costs (e.g., storage), and medical errors to improve data availability and sharing, etc. An exemplary successful implementation of EHR system in the United States is the Veterans Administration health-care system, with over 155 hospitals and 800 clinics. It is one of the largest integrated health-care information systems worldwide and has been using a single EHR system for years [28]. Despite all the promising factors, EHR systems are not adopted by the majority of health-care systems. Statistical results [29, 30] show very low actual adoption rate of EHR in the U.S. medical systems.

Among all the barriers to the implementation of EHR systems, privacy and security concerns on patients' medical records are arguably most dominating [28, 31]. EHR will inevitably be stored in remote servers (e.g., monitoring center and primary health-care provider) and exchanged over the Internet for cooperative treatment, emergency response, clinical research, etc., and thus are subject to theft and security breaches. The Health Insurance Portability and Accountability

Act (HIPAA) in the United States was established to regulate EHR-related operations. Privacy issues are particularly not addressed adequately at the technical level. Therefore, 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 health-care providers [5]. The implementation of the standardization or strategy will undoubtedly be relying on technical details, which are rarely studied in the research realm leaving numerous research opportunities.

As the need for technical details, i.e., the cryptographic realization of secure EHR systems becomes more clear and urgent; a few recent works followed this line of research, including cryptographic key management schemes, role-based access control schemes, anonymous authentication schemes, etc. These works mostly focus on a single problem or aspect of the system and thus would fail when taking other aspects and objectives into consideration. Technical solutions for the assurance of privacy and system-wise security in e-health care are yet to come.

27.3. PRIVACY AND SECURITY IN E-HEALTH CARE

We provide a non-exhaustive list of privacy and security issues that concern patients and will serve as requirements/objectives in future e-health-care system design. We also discuss the suitable cryptographic techniques for solving these issues.

27.3.1. Privacy

Privacy is of paramount importance in e-health care because the illegal disclosure and improper use of EHR can cause legal disputes and damaging consequences to people's lives. For example, an employer may decide not to hire people with psychological issues, an insurance company may refuse to provide life insurance knowing the disease history of a patient, people with certain types of disease may be discriminated by the health-care provider, health conditions of the elderly could be revealed to their families disobeying their willingness, and so forth [28, 31].

Privacy in e-health-care environment comprises anonymity and unlinkability requirements [28]. Anonymity is required when the identifying information in the EHR need be hidden from certain parties, i.e., the EHR cannot be associated with a particular patient. These parties include insurance providers, researchers, some management staff, and any related personnel who has no appropriate access privileges. On the other hand, primary health-care providers including physicians and nurses, delegated health-care providers, and emergency medical technicians (EMTs) [32] should be able to view such information in order to carry out proper treatment. In addition, the identity of a patient can be deduced from the received medical data if the patient's device (e.g., home PC, PDA) transmitting the data can be identified.

Unlinkability indicates that multiple EHRs cannot be linked to a same owner. This requirement is necessary because it prevents the profiling of a patient by insurance companies or central servers that store patient data [28]. The insurance companies may benefit from learning more information than that is allowed by the patient, through exploiting the linkage among EHRs. The monitor centers, either independent or within a hospital, offer services and storage to patients under home or critical care. The monitored data can then be retrieved by the primary physician for health evaluation or by the emergency medical technicians for ambulatory treatment. The unlinkability requirement applies to the storage servers (i.e., the administrators) of the monitor center, under the curious-but-honest assumption meaning that the servers will attempt to learn the privacy of the patient but will not launch attacks on the stored EHRs (e.g., deletion, modification, bogus injection, and irresponsive to retrieval requests). It is apparent that anonymity is a prerequisite for unlinkability, because identifying information renders EHRs linkable.

One can use data anonymization techniques to remove identifying information and achieve the anonymity of EHRs [28]. The anonymization can be performed by the patient or the primary physician to allow sharing of EHRs with insurance companies, researchers, cooperative health-care providers, etc., without compromising privacy. Since data anonymization techniques fail to ensure anonymity when the device transmitting data is identified, the aforementioned anonymous communication substrate for location privacy will also be needed here. Furthermore, most commonly, the device will be required to authenticate with the storage servers of the monitor center before transmitting health data and to prevent users who are not subscribed to the monitor services from abusing the servers. Since authentication relies on public key infrastructure (PKI), the public key of the device must be anonymous, which can be realized by adopting anonymous credential systems. At this point, it is clear that the anonymity requirement is multifaceted, the negligence of which will cause failures in the anonymity guarantee. Anonymization techniques can be leveraged to achieve unlinkability, because the removal of common identifiers in the EHRs results in ambiguity. Encryption can also be used which encrypts EHRs and produces ciphertexts that appear random, and thus unlinkable. More discussions on suitable encryption schemes for e-health-care can be found in the confidentiality requirement below.

27.3.2. Access Control

Access control is in charge of who can access the patient's EHRs and which part(s) of the data can be accessed, to ensure that only authorized parties can gain access to authorized data [28]. This requirement is in accordance with the HIPAA regulation that patient authorizations will be required to use and disclose information for purposes other than treatment and payment [1]. Basically, the identifying information (or protected health information, PHI) is necessary for treatment and payment, where authorization can be exempted. In all other cases, patients have the right to permit the use and disclosure of their EHRs, and hence the access control should be patient-centric. Access control is an intrinsic issue due to the various types of personnel, medical, or non-medical, involved in the interactions between patients and the health-care systems.

Role-based access control is the de facto mechanism to deal with authorizations in health-care systems, where the roles (e.g., physician, nurse, emergency medical technician, insurance provider, pharmacist, and cashier) and their associated access rights can be defined and specified. It greatly simplifies the control task in that access is determined and granted for each group of people but not individually [28]. Translating to cryptographic details, the public key used for authentication and secure communications will be constructed from the descriptive string of a role, as opposed to that of an identity. Fairly often, patients need to be referred to specialists for the examination and treatment of certain health conditions, upon the primary physician's discretion. The specialists will, therefore, have temporary access to the entire or partial EHR, during the course of treatment. Temporary access implies the need for potentially frequent assignment/revocation of the roles, which can be fulfilled by means of delegation. Delegation refers to the primary physician delegating access right to another physician and specifying the associated validity period. Delegation can also be role based, where the primary physician delegates his/her role to another physician, and revokes the role upon the termination of treatment. Depending on the policies and applications, onward delegation may be allowed in which the delegated physician can further delegate another physician. The depth of the delegation chain will normally be defined by the primary physician. In addition, delegation can be realized through proxy signature/certificate and XML-based approaches.

The role-based approach solves the problem of who has access to the EHR. However, it alone cannot provide granularity in EHR access (i.e., what portion(s) of the EHR a particular role has access to), which requires additional mechanisms such as anonymization and encryption.

Anonymization is indispensable for data mining performed by parties such as researchers and insurance providers who possess access rights only to the de-identifying or sensitive information of EHRs. The anonymization may be carried out by the patient as the EHR owner or by the primary physician who can act on the patient's behalf. Encryption is another option and is more precise in restricting the access. The patient and primary physician can simply encrypt the EHR portion(s) to be accessed by a role leveraging role-based encryption (i.e., the public key used for encryption indicates a role). This manifests another merit of role-based technique, being that the encryption can be performed in advance without knowing the identity of the potential recipient.

27.3.3. Authentication

Authentication is a prerequisite for secure operations or tasks, since the communicating parties need to ascertain the legitimacy and authenticity of each other [28]. Hence, authentication procedure should be executed as the first step of almost all communications in e-health-care systems. For instance, authentication takes place as patients transfer data to the monitor center or request test results from the primary physician, primary physician retrieves the data for health evaluation or delegates other physicians, researchers request EHRs for statistical studies, and so on.

Authentication in the e-health-care context relies on public key infrastructure (PKI), where a cryptographic public/private key pair is imperative. Assigning key pairs for authentication in e-health-care systems is challenging, in that most of the aforementioned communications occur in an inter-domain fashion. The domain is defined such that a trusted authority can be easily established to assign key pairs for every entity in the domain, facilitating intra-domain authentication. In general, a hospital, clinic, insurance company, research organization, monitor center, or ambulatory treatment center can be considered as a domain, where a server may be designated to assign key pairs for the employees with common affiliation. Moreover, patients having business or research relationships with a domain will possess a key pair for that domain. In the inter-domain authentication scenario, communications involve two independent domains, the key pairs of which cannot mutually authenticate. As a result, a common certificate authority (CA) need be found in certificate-based PKI or the hierarchical identity-based cryptosystem (HIDC) need be adopted in identity-based PKI. Nevertheless, the certificate-based PKI is inappropriate in the e-health-care context because it renders the role-based techniques introduced in the previous section infeasible. We will later demonstrate the possibility of finding common trust for inter-domain authentication in e-healthcare leveraging HIDC.

27.3.4. Confidentiality and Integrity

Confidentiality and integrity are with respect to EHRs. In particular, confidentiality requires that the entire or partial EHR, depending on the application and patient's requirement, is viewable only to parties with proper authorizations (i.e., decryption keys) and is achieved by encryption primitives [28]. Encryption was mentioned as one of the techniques (another being anonymization) to be used with role-based approaches for fine-grained access control. It is clear that the major difference between encryption and anonymization lies in the assurance of confidentiality, as the remaining information in the EHRs after anonymization is still viewable. Confidentiality is indispensable when it is undesirable to reveal sensitive information in the EHRs, even when this information is not identifying. For example, patients with certain types of disease may feel embarrassing to disclose related information for any use other than necessary treatment.

Symmetric or public key encryption can be used, where the former requires a shared secret key between the encrypting and decrypting party, and the latter can use the public/private key pairs assigned for authentication. Apparently, public key encryption is the suitable solution in e-health-care context to provide basis for role-based techniques. The basic encryption schemes are most useful for real-time communications. Frequently, in e-health-care applications, the encryption of medical data will take place before the actual decryption and review, e.g., the EHRs are stored by the primary health-care provider for future reference, the monitored health data (raw or processed) are outsourced to central servers for later evaluation by the primary physician. Furthermore, the retrieval of medical data should be highly specific and efficient, in that only the set of EHRs relevant to the retrieval (out of a potentially large pool of EHRs) need be returned. Considering this feature of medical data retrieval, special public key encryption schemes designed for efficient retrieval purposes should be adopted instead of the basic schemes. Public key encryption with keyword search (PEKS), or simply, searchable public-key encryption, is a desirable candidate, which supports the functionality of role-based (basic) public key encryption while facilitating efficient search for data of interest.

Integrity of EHRs needs to be ensured so that illegal alteration of the original EHRs can be detected by future reviewers. It is critical to satisfy the integrity requirement in e-health-care systems, since illegal modification of the EHRs (either maliciously or erroneously) may result in life-threatening consequences [28]. Traditionally, integrity is guaranteed by public key–based digital signature or symmetric key–based message authentication code. The former is expected to be the dominant technique for e-health-care applications, and the latter is useful when the patient shares a secret with his/her family for EHR access. Another popular approach for integrity guarantee is the watermarking technique applied in medical information security, to assure the integrity and authenticity of the medical data (e.g., images, texts, videos, audio, etc.) in which the watermark is embedded. The challenge in the watermarking technique is to achieve the security objectives, and meanwhile to yield minimal impact on the quality of the original data for diagnosis. Watermarking can be used to integrity assurance, so long as the distortion is acceptable for the purpose of the application. Otherwise, the cryptographic approaches mentioned above should be leveraged to avoid medical incidents caused by inaccurate diagnosis or misdiagnosis.

27.3.5. Others

Other security requirements including availability and accountability need also be satisfied [28]. The most common attack on availability is Denial of Service (DoS) or Distributed Denial of Service (DDoS) in distributed systems. The attacker may flood the servers storing EHRs with continuous bogus authentication requests (recall that authentication is need for all secure communications in e-health care) to cause irresponsiveness at the server hindering critical data retrieval. In the wireless transmissions of medical data from the PDA to the monitor center, the attacker can launch jamming attack rendering the wireless channel saturated and unavailable, and thus cause delay in the delivery of critical data. DoS and jamming attacks are very difficult to thwart. The best solution so far is to alleviate the impact of such attacks and is application specific taking into account the features of the application and its objectives.

Accountability, also referred to as non-repudiation or traceability, provides the possibility to trace and identify the party that misbehaves. The definition of misbehavior is application specific and consists of a wide range of actions violating the regulation, policy, or security requirements. Misbehavior can include illegal disclosure of EHRs, abuse of access rights for illegitimate purposes, unauthorized modification to EHRs, collusion of insiders (e.g., physicians and insurance companies) for monetary gains, etc. To enable accountability and discourage misbehavior, audit trail and cryptography (i.e., digital signature) should be used in combination. Audit trail is available in many systems as the data logger to record transactions and events occurred, for statistics, quality of service, or security purposes. In e-health-care systems, audit trail functionality should be provided whenever digital signature is not required, to trace the source that breaks the rules and possibly also causes damages.

27.4. STATE OF THE ART DESIGN FOR HEALTH INFORMATION PRIVACY AND SHARING (HIPS)

27.4.1. Entities and Definitions

The following entities are involved in the HIPS system. *Patient* is the user of HIPS and is referred to as the combination of a person and his/her computing facilities (personal desktop, or any wireless-enabled portable devices [6, 33–35]). *Physician* including *Delegator* and *Delegatee* denotes health-care professionals such as doctors, nurses, pharmacists, and any other individuals licensed to provide health-care services. Similar to patient, physician refers to a person and his/her computing facilities. In the information sharing procedure, we will refer to the physician who shares his/her patients' information as the delegator, and the physician with whom the information is shared as the delegatee. Other than the information sharing procedure, we simply use physician to denote the primary caregiver (corresponding to the delegator in information sharing). *Family* represents any person(s) of the patient's trust including parents, children, spouse, relatives, or close friends, who will possess the secrets for retrieving the patient's PHI in emergencies and are assumed unlikely to compromise the patient's privacy in any case. *P-device* (private 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. It must be pervasive for patients with high risks to encounter emergencies.

S-server (storage server) is provided by each hospital/clinic to store the patient's PHI, which is subject to compromise and is generally not trusted by patients. The S-server can be assumed honest-but-curious meaning that it will not maliciously delete patients' PHI for gaining nothing. *D-server* (delegation server), equipped with decision making functionalities and possessing access to the privacy and security policies of the delegatee's organization, plays the role of proxy signer (or mediator) in information sharing and is in charge of properly assigning delegatees on the delegator's behalf. *A-server* (authentication server) is a trusted server run by the government (e.g., NHIN and HHS) and placed in local offices, responsible 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.

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), containing both sensitive and non-sensitive information. It also refers to information with respect to which there is a reasonable basis to believe the information can be used to identify the individual [36]. We are interested in the electronic PHI in our EHR system, HIPS. We use PHI to represent patient-controlled health information that is mainly for common or emergency treatment. SHI, the shared health information, refers to the patients' health information shared between organizations in collaboration, where patients themselves are not involved. SHI in the case of refereed treatment is identical in content to PHI. However, in the case of collaborative clinical research, sensitive information is usually removed through de-anonymization and SHI is different from PHI.

27.4.2. Security Requirements

It is crucial to design HIPS against a set of predefined security requirements, indicating the main functional objectives HIPS should satisfy as a secure system. Provided the application scenario with stringent privacy demands, in our secure system HIPS, we will define and address the following security requirements [31, 37]:

1. *Privacy*: HIPS achieves privacy if patients' PHI can *only* be accessed by authorized physicians for legitimate reasons (i.e., treatment, payment, health-care operations [1]), and no one

except the family and P-device can link the stored PHI files to a particular patient.

2. *Fail-Open*: We say that HIPS system is fail-open if the system provides backup mechanisms to successfully retrieve patients' PHI in case of emergency while preserving the above privacy requirement.
3. *Access Control (Authorization)*: HIPS realizes access control if no physicians other than the authorized can gain access to the patient's PHI or the SHI.
4. *Accountability*: HIPS meets the accountability goal if the physician who discloses the patient's PHI and the SHI other than legitimate reasons is traceable and held responsible in case of emergency and information sharing, respectively. We implicitly assume that when the patient is physically competent to retrieve the PHI (i.e., not in emergency), he/she will know the source of the PHI leakage by recalling which physician(s) recently treated him.
5. *Minimum-Privilege Delegation*: Our system achieves minimum-privilege delegation if the delegator is able to specify which data portions of SHI can be accessed by the delegatee, even if these data portions belong to a same document as those that cannot be accessed by the delegatee.
6. *Adaptability*: Our system meets the adaptability goal if the change of status or availability of a delegatee does not require the intervention of the delegator to restart the information sharing procedure, nor cause the interruption of the procedure in any way. In other words, the changes should be transparent to the delegator.
7. *Dynamic Revocation*: We say that our system guarantees dynamic revocation if the system provides mechanism for the delegator to revoke delegated rights at any time.
8. *Availability*: The availability requirement states that the authorized physician must be able to obtain PHI and SHI stored anywhere in the health-care architecture.
9. *Authenticity*: Authenticity indicates that any entities involved in HIPS communications must be able to successfully authenticate or verify the identity of each other, even if such authentication is cross-domain.
10. *Confidentiality*: Confidentiality requires that the contents of PHI and SHI are not learned by any passive eavesdroppers or active attackers, which fundamentally guarantees patient privacy specified in the privacy requirement. Furthermore, message exchanges involving secret information are subject to confidentiality requirement as well.
11. *Data Integrity*: HIPS guarantees that the stored PHI or SHI 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.

27.4.3. System Architecture

Consider the application scenario in our HIPS system shown in Figure 27-1, where all links are bidirectional and the bracketed numbers indicate major events or exchanged messages. In general, the physician has only physical contacts with all entities in a patient local area network (LAN), denoted by a double solid line from the physician, Dr. White in Figure 27-1, to a 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 cell phone.

The internal links of the hospital/clinic network and the patient LAN are often high-speed wired links. The patient interacts with the family and P-device to assign privilege (i.e., secret keys)

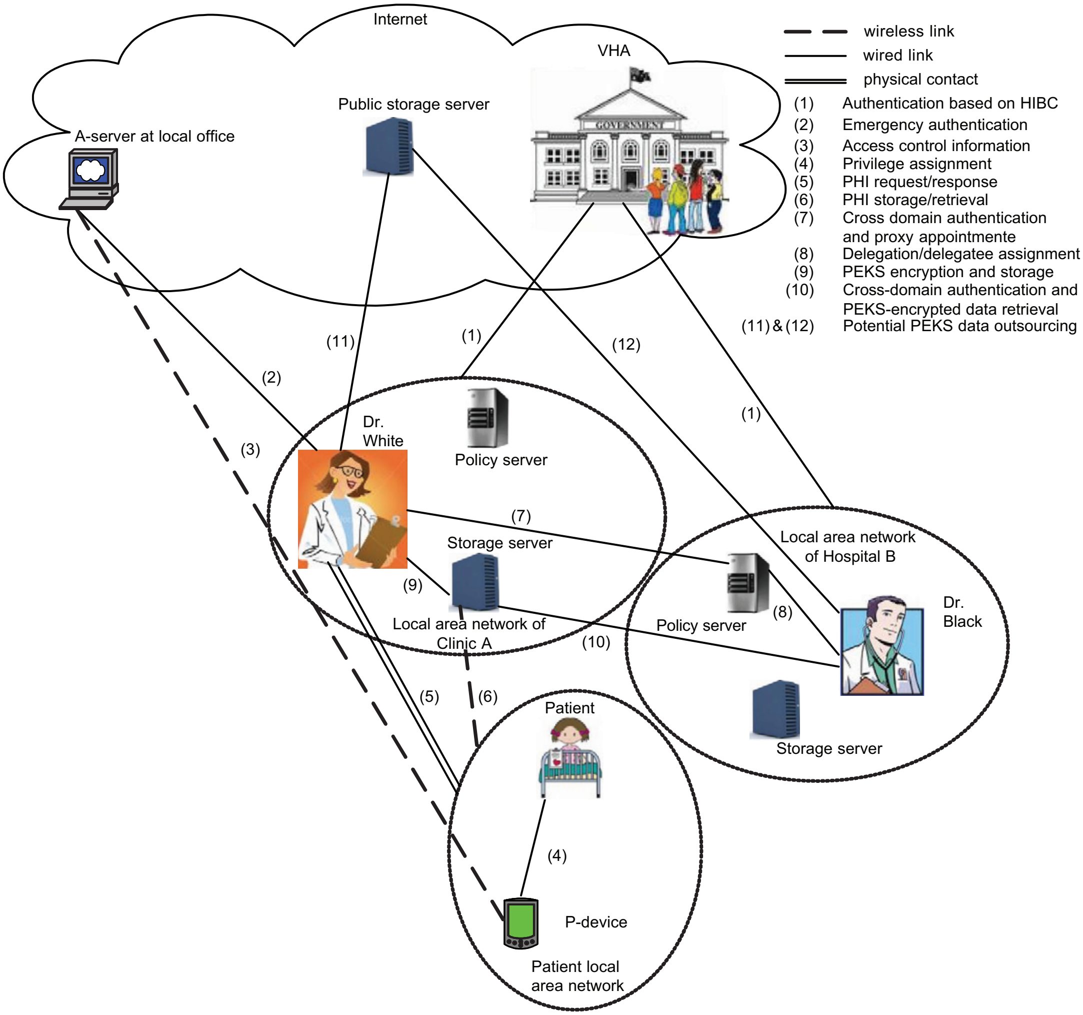
Internet
VHA
Public storage server
A-server at local office
— — wireless link
—— wired link
═══ physical contact
(1) Authentication based on HIBC
(2) Emergency authentication
(3) Access control information
(4) Privilege assignment
(5) PHI request/response
(6) PHI storage/retrieval
(7) Cross domain authentication and proxy appointmente
(8) Delegation/delegatee assignment
(9) PEKS encryption and storage
(10) Cross-domain authentication and PEKS-encrypted data retrieval
(11)&(12) Potential PEKS data outsourcing
(11)
(1)
(12)
(2)
(3)
(1)
Dr. White
Policy server
(7)
Storage server
(9)
Local area network of Clinic A
Local area network of Hospital B
Policy server
(8)
Dr. Black
(10)
(5)
(6)
Patient
Storage server
(4)
P-device
Patient local area network

FIGURE 27-1 System architecture of HIPS.

that will be used for retrieving the patient's PHI in emergencies.

The Veterans Health Administration (VHA) serves as the parent of Clinic A and Hospital B in the hierarchy shown in Figure 27-2 for authentication and key management. The delegator Dr. White and S-server will be engaged in SHI storage. A public S-server is also depicted in Figure 27-1 to show the potential outsourcing of the encrypted SHI, enabling distributed health information storage/retrieval. The PEKS (searchable public key encryption) primitive [38–40] used for encrypting SHI ensures that the content of the stored data will not be recovered by the S-server (or any other third-party entities except the intended delegatee, Dr. Black), regardless of the location of the server in the health-care hierarchy. Note that although we use the S-server located within Clinic A for demonstration, any other S-server drawn in Figure 27-1 can be used instead. Revocation takes place in Event (10) in between cross-domain authentication and PEKS-encrypted SHI retrieval, which is not shown in the figure. Also not included in the figure is the

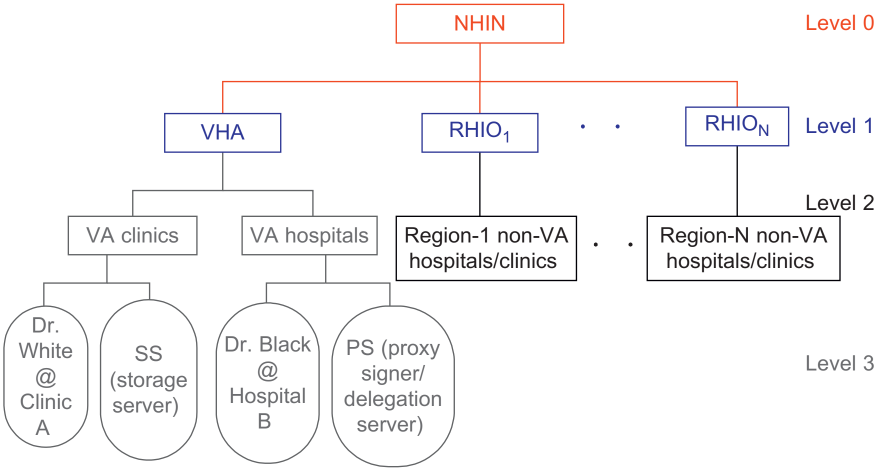

FIGURE 27-2 Logic diagram of HIPS system hierarchy.

consultation to the D-server in Event (10). Intuitively, the S-server should mainly be responsible for storage, with authentication and revocation capabilities left up to the D-server. These design details can be easily handled and will be the policy specification issue in the implementation of HIPS as a part of our proposed research.

All entities in Figure 27-1 need to individually maintain audit trails for recording interaction histories (e.g., authentication, proxy signing, PHI/SHI storage, retrieval, revocation, etc.) with other entities. Audit trails serving as proofs will be critical for audit once disputes arise regarding serious issues such as abuse of permissions, illegal attempts of access, improper disclosure of patients' health information, etc., and will be the key technique in satisfying the accountability requirement.

27.4.4. Establishing Trust: Authentication and Key Management

Establishing trust is fundamental to properly ensuring other security procedures and mechanisms since it essentially sets up a secure backbone for all future authentication, confidential and integrity-protected message exchange, and key management. This procedure thereby ensures the security requirements of authenticity, confidentiality, data integrity, and availability.

United States has one of the largest integrated health-care delivery systems based on an EHR information system VistA, which is administered by Veterans Health Administration (VHA), the health-care/medical organization of U.S. Department of Veterans Affairs (VA). Consequently, federal department VA can act as the common ancestor of all VA health-care providers (e.g., VA hospitals, VA clinics, etc.) forming a hierarchy for enabling cross-domain authentication among these providers.

Outside the VA system, however, there was lack of adoption of EHR systems or incompatibility of EHR software between vendors [2]. As a result, the Office of the National Coordinator was established within the U.S. Department of Health and Human Services (HHS), striving to build Nationwide Health Information Network (NHIN). Within NHIN, Regional Health Information Organizations (RHIOs) have been established in many States in order to promote the sharing of health information [2]. It provides a platform for cross-domain authentication among the non-VA health-care providers, as well as between VA and non-VA providers by incorporating VA providers as participants of NHIN [41].

The above example indicates the possibilities of finding common trust for information sharing in the health-care system, supporting the proposed solution that leverages HIBC for cross-domain authentication. In addition, Lim and Paterson [42] showed the feasibility of implementing HIBC in practice, and the performance superiority of HIBC over certificate-based cryptosystem (i.e., RSA) in terms of communication costs. Another advantage of HIBC is the capability of directly authenticating a public key. If the certificate-based hierarchical cryptosystem is deployed, the verification of a certificate usually involves verifying all certificates along the certification path until a trusted certification authority is reached [31].

We apply the domain initialization of HIBC [43]. According to the above discussion, we let NHIN locate at level 0 of the hierarchy. VHA and RHIOs, and their affiliated health-care providers, are located at level 1 and level 2, respectively, as shown in Figure 27-2. NHIN being the root PKG (public key generator) performs the following domain initialization algorithm when HIPS is bootstrapped, where P_0 is a generator of G_1 [37].

1. Input security parameter $\xi \in Z^+$ into domain parameter generator $\mathcal{PG}$ and output the parameter tuple $(q, G_1, G_2, e, P_0, H_1)$.
2. Randomly select a domain master secret $S_0 \in Z_q^*$ and calculate the domain public key $\overline{P_{pub}} = S_0 P_0$.

NHIN publishes the domain parameters $(p, G_1, G_2, e, P_0, H_1, \overline{P_{pub}})$ and maintains S_0 confidential. The setup for an entity CH_j at level j, $j \in \{1, 2\}$ in our scenario is performed by CH_j's immediate ancestor (or parent) at level $j-1$ as:

1. Compute $PKT_j = H_1(ID_1, \ldots, ID_j)$;
2. Compute CH_j's private keys $\psi_j = \psi_{j-1} + S_{j-1} PKT_j = \sum_{i=1}^{j} S_{i-1} PKT_i$;
3. Distribute $QT = \{Q_l : 1 \le l < j\}$ to CH_j, where $Q_l = S_l P_0$.

In the above private key assignment, $IDT_i = (ID_1, \ldots, ID_i)$ for $1 \le i \le j$ is the ID tuple of CH_j's ancestor at level i and PKT_i is the corresponding public key tuple. The private key ψ_j is generated for cross-domain authentication, where S_{j-1} is the parent's secret information. Due to the hardness of discrete logarithm problem (DLP), it is intractable to solve for S_{j-1} given any private key calculated from it with non-negligible probability.

27.4.5. Protecting Patient Privacy

This procedure guarantees the privacy requirement by providing privacy protection for patient during the PHI storage and future retrieval. It is executed by the patient whenever the PHI is created, updated, or modified (e.g., after diagnosis or tests). The patient breaks the PHI into files for different categories of health information (e.g., allergy lists, drug history, X-ray data, surgeries, etc.).

The files are encrypted using any semantically secure symmetric key encryption and stored with a secure index (SI) in the S-server. The construction of SI can be based on searchable symmetric encryption [44].

From the visited hospital, the patient can obtain the URL link to the S-server, and a temporary HIBC public/private key pair, based on which the patient can generate a valid pseudonymous key pair $\widetilde{PKT}_p/\tilde{\psi}_p$ (cf. pseudonym self-generation in Ref. [45]), 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., the encrypted files and SI) to S-server [31]. Furthermore, the patient can self-generate multiple key pairs $\widetilde{PKT}_p/\tilde{\psi}_p$ for different searches, so that his/her successive activities will not be linked under the same $\widetilde{PKT}_p$.

27.4.6. Controlling Access to Patients' Health Records

This procedure is applied to the physician's access to PHI, or to the delegatee's access to SHI, fulfilling the access control (authorization) requirement.

Controlling the Access to Personal Health Information. On subsequent visits to the hospital, the patient will be asked for the PHI relevant to the treatment being sought. The PHI retrieval can be efficiently performed as a result of the searchable symmetric encryption technique.

The access control is executed between the patient and S-server. The patient transmits ($\widetilde{PKT}_p$, SI, $TD(kw)$) to S-server, where $TD(kw)$ is the trapdoor for keyword kw and all other parameters have been defined before. S-server executes an algorithm SEARCH locally and outputs $\Lambda(kw)$, the set of encrypted files containing kw. The patient decrypts the received $\Lambda(kw)$ on his/her cell phone and sends the plaintext PHI to the physician [31]. The cell phone would suffice due to the low complexity of the retrieval protocol. The key point of adopting the keyword search is the small number of files (instead of the entire file collection) returned to the patient. 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.

Controlling Access to Shared Health Information. This procedure assures the security requirement of minimum-privilege delegation, in addition to access control. Uncertain in advance of which doctor will be selected by D-server as the delegatee, the delegator encrypts the intended data using the role-based descriptive identity string IDT_R (or more accurately, the corresponding public key PKT_R) representing the delegatee's role in the hierarchy, for which only an authentic entity in that role at the desired organization has the corresponding private key ψ_R and secret key S_R. The delegator Dr. White stores the PEKS-encrypted SHI by uploading ($\mathcal{HIBE}_{PKT_R}(PatientData) \| PEKS(KW)$, t_1, $\mathcal{HMAC}_\nu(\mathcal{HIBE} \| PEKS \| t_1)$) on S-server for future searches, where $PatientData$ denotes the PEKS-encrypted SHI and $HIBE$ represents hierarchical ID-based encryption. The pre-shared secret key ν is assumed to exist between Dr. White and S-server in the same domain or can be easily established otherwise. The keyword KW, different from kw in the PHI case, can be the delegator's identity, date/time the encryption is created, etc., or any combination of them [40].

When Dr. Black needs to access the intended patient data, he/she first authenticates with S-server to prove appropriate access permissions by transmitting $(IDT_R, TD(KW), t_2, \mathcal{HIDS}_{\psi_{D-server}, S_{D-server}}, \mathcal{HIDS}_{\psi_{White}, S_{White}}, \mathcal{HIDS}_{\psi_R, S_R}(PKT_R \| TD(KW) \| \mathcal{HIDS}_{\psi_{White}, S_{White}} \| \mathcal{HIDS}_{\psi_{D-server}, S_{D-server}} \| t_2))$, leveraging the proxy signatures obtained from the proxy signer D-server, where $TD(KW)$ is the trapdoor computed by Dr. Black for searching KW. The S-server essentially performs the same proxy signature verifications as the delegatee, to assure the authenticity and permissions of both the proxy signer and the delegatee. The signature $\mathcal{HIDS}_{\psi_R, S_R}$ is present for S-server to verify the proper role R Dr. Black is claiming (recall that PKT_{Black} is included in the proxy signature to enable such verification, mainly for later audit once disputes arise). The proxy signature verification also entails S-server checking Dr. Black's revocation status. If not revoked, Dr. Black can obtain the PEKS-encrypted SHI, by receiving $\mathcal{HIBE}_{PKT_R}(PatientData)$ from S-server.

27.4.7. Emergency Health Information Retrieval

This procedure 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*.

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. *Family* is equipped with all necessary information to retrieval PHI in emergencies (the same procedure used *patient* in the normal case).

The essence of family-based approach captures the security factor of "someone you trust" [46], 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 if the family based approach fails [31].

P-Device-Based Approach. As health-care 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 HIPS, 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.

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/her hospital for work and sign out when he/she leaves, so that the list of "today's on-duty physicians" of each hospital can be published online for A-server to check against [31]. The detailed PHI retrieval using P-device can be found in Ref. [31].

27.4.8. Sharing Information for Health Care and Research

Delegation. Our design features role-centered delegation, which essentially fulfills the adaptability requirement of HIPS by delegating ageneral role instead of a specific entity. After cross-domain authentication is successfully executed using the keys assigned, Dr. White can delegate access rights to his/her patient data to Dr. Black with whom trust could be established indirectly via the D-server (see later this section). However, directly delegating a specific person does not scale with dynamic changes. As in the example above, when Dr. Black becomes unavailable during delegation or the cooperation afterwards, Dr. Brown has to be delegated again to continue Dr. Black's duty. A feasible solution yielding dynamics would be to delegate the access rights to a role instead of a person in that role, e.g., delegating a rheumatologist (both Dr. Black and Dr. Brown are rheumatologists) at Hospital B.

We recognize that there are two cases in practice concerning role-based delegation [37]: (a) the same role exists in the organization of both the delegator and the delegatee and (b) the role to be delegated is delegatee-specific and has no mapping in the delegator's organization. Case (a) is possible when the two health-care provider organizations are conducting cooperative research involving shared data, and the two groups of researchers hold similar positions in their respective groups. Case (b) occurs as daily routine, where the delegator frequently refers patients to specialists that are unavailable at the delegator's organization. Case (a) is easily coped with since Dr. White, the delegator, can specify rules and permissions for Dr. Black, the delegatee, according to the policies of Dr. White's organization (Clinic A) on Dr. Black's role (i.e., rheumatologist). Dr. White then digitally signs the rules and permissions, which will be issued to Dr. Black as evidence of delegation. The solution to Case

(b) is not straightforward and deserves elaboration. A naive approach would be to let the delegator learn relevant policies in the delegatee's organization. For one thing, such an approach does not scale since the delegator may eventually have to learn all policies associated with all possible roles. Furthermore, policy learning is by no means an easy task [15] which will incur complexity in the delegation process, especially at the delegator's side. For another, policies specifying permissions to a role are usually for internal uses only and are hence confidential information not to be disclosed to any other organizations. A feasible solution would be for the delegator Dr. White to appoint the policy server (or role-permission server) PS of the delegatee's organization, which acts on the delegator's behalf to select a qualified delegatee Dr. Black. Note that the design of policy server avoids the complexity involved in delegation-chaining, where Dr. Black can further delegate someone in his/her role due to various reasons (e.g., availability). As a result, the delegator from Clinic A will only need to carry out cross-domain authentication and proxy appointment with the policy server, as illustrated in Figure 27-1.

The delegation procedure is described as follows, immediately after the cross-domain authentication between Dr. White and PS with HP_1, HP_2 replaced by Dr. White, PS, respectively) [37]:

1. $White \to PS$: IDT_{White}, W, $\mathcal{HIDS}_{\psi_{White}, s_{White}}(00 \parallel W \parallel PKT_{PS} \parallel PKT_R)$, t_4, $\mathcal{HMAC}_{\zeta}(PKT_{White} \parallel W \parallel \mathcal{HIDS}_{\psi_{White}, s_{White}} \parallel t_4)$;
2. $PS \to Black$: IDT_{PS}, W, m $\mathcal{HIDS}_{\psi_{White}, s_{White}}$, $\mathcal{HIDS}_{\psi_{PS}, s_{PS}}(01 \parallel PKT_{White} \parallel m)$, t_5, $\mathcal{HMAC}_{\varsigma}(PKT_{PS} \parallel W \parallel \mathcal{HIDS}_{\psi_{White}, s_{White}} \parallel \mathcal{HIDS}_{\psi_{PS}, s_{PS}} \parallel t_5)$;
3. $Black \to PS$: ID_{Black}, PUB_R, t_6, $\mathcal{HMAC}_{\varsigma}(PK_{Black} \parallel PUB_R \parallel t_6)$;
4. $PS \to White$: IDT_{PS}, PUB_R, t_7, $\mathcal{HMAC}_{\varsigma}(PKT_{PS} \parallel PUB_R \parallel t_7)$,

where $PK_{Black}/\lambda_{Black}$, ς, $\mathcal{HMAC}$ denote the standard ID-based public/private key pair, the shared secret key between PS and Dr. Black, and the keyed-hash message authentication code for data integrity check, respectively, with $PK_{Black} = \lambda_{Black} H_1(ID_{Black})$. The role-based public key/ID string PKT_R/IDT_R describes the role of the delegatee Dr. Black and is assigned together with PKT_{Black}/IDT_{Black} at domain initialization. The role-based credentials can be deduced by anyone in the hierarchy since they are formed the same way as PKT_{Black}/IDT_{Black}, except with a general role "Rheumatologist" replacing a specific name Dr. Black. The warrant W containing the expiry date of the appointed proxy PS is used for restricting PS's signing rights as the proxy signer. In general, the proxy signing rights expire as the role-based delegation terminates. The numeric strings prepended in the above signed messages are necessary for provably secure proxy-signature-based delegation [17]. Since PS and Dr. Black are in the same organization or management domain, we assume the pre-shared secret keys between these two entities. However, the hierarchical ID-based signature $\mathcal{HIDS}_{\psi_{PS}, s_{PS}}$ is used instead of the standard ID-based signature due to the future cross-domain proxy signature verification, which will be explained in a later section. Refer to Refs. [45] and [47] for standard ID-based domain initialization and the instantiation of $\mathcal{IBS}$, respectively. Upon receiving the delegation message m, $m := (PKT_{Black} \parallel PKT_R)$, the delegatee Dr. Black can verify the proxy signatures including the delegator Dr. White's signature $\mathcal{HIDS}_{\psi_{White}, s_{White}}$ on the warrant and the proxy signer PS's signature $\mathcal{HIDS}_{\psi_{PS}, s_{PS}}$ on the delegation message, by performing $\mathcal{HIDV}_{PKT_{White}}(00 \| W \| PKT_{PS} \| PKT_R, \mathcal{HIDS}_{\psi_{White}, s_{White}})$ and $\mathcal{HIDV}_{PKT_{PS}}(01 \| PKT_{White} \| m, \mathcal{HIDS}_{\psi_{PS}, s_{PS}})$, respectively. The signature verification algorithms $\mathcal{HIDV}_{PKT_{White}}$ and $\mathcal{HIDV}_{PKT_{PS}}$ correspond to $\mathcal{HIDS}_{\psi_{White}, s_{White}}$ and $\mathcal{HIDS}_{\psi_{PS}, s_{PS}}$, respectively. Upon successful verification, the delegatee Dr. Black returns to the policy server PS an additional public key PUB_R, which is necessary for fine-grained access control and is eventually returned to the delegator Dr. White. Note that Steps 3 and 4 can be eliminated if the fine-grained access control is opted out, due to the unnecessary employment of this advanced control mechanism in some application scenarios.

Fine-Grained Access Control. The delegation procedure discussed in the previous section can be regarded as general access control, in that only those who are delegated proper rights can access private patient data. However, the delegation of rights is role-based, which indicates that the delegatee is granted *all* permissions associated with a particular role, possibly including permissions beyond the access of intended patient data. Such coarse-grained access control is insufficient especially in the case where there is no role mapping between organizations (i.e., Case (b) above), since the delegator's organization has to rely entirely on the policies of the delegatee's organization. Although the policy server (i.e., the proxy signer) of the delegatee's organization can be held guilty if its behavior is not compliant with the policies by inspecting the audit trails, such an approach is reactive rather than preventive. To proactively gather more control over the portion of patient data to be accessed by the delegatee, the delegator can place an extra layer of fine-grained control to limit the access to only minimum necessary amount of data. Note that the fine-grained access control is optional, provided that delegation is in place as the general access control mechanism. However, the fine-grained control is highly recommended for handling sensitive data such as those in EHR context.

The fine-grained access control is based on PEKS, the searchable public-key encryption. In our EHR environment, the delegator selectively encrypts those data intended for access under the delegatee's public key and stores the encrypted data on a storage server at the delegator's organization. Note that the storage server is subject to corruption and other attacks. Using PEKS, the encrypted patient data can only be accessed by the delegatee as the intended receiver.

Uncertain in advance of which doctor will be selected by the policy server as the delegatee, the delegator encrypts the intended data using the role-based descriptive identity string IDT_R (with the corresponding public key PKT_R) representing the delegatee's role in the hierarchy, for which only an authentic entity in that role at the desired organization has the corresponding private key ψ_R and secret key S_R. The delegator Dr. White stores the encrypted patient data in the storage server SS for future searches as [31]:

$White \rightarrow SS$: ID_{White}, $\mathcal{HIBE}_{PKT_R}(PatientData) \parallel PEKS_\sigma(KW)$, t_8, $\mathcal{HMAC}_\nu(PK_{White} \parallel \mathcal{IBE} \parallel PEKS_\sigma \parallel t_8)$,

where $PEKS_\sigma(KW) = (\sigma P_0, H_3(e(H_2(KW), PUB_R)^\sigma))$ is the searchable public key encryption with $\tau, \sigma \in_R Z_q^*$ chosen by Dr. White and $PUB_R = \tau P_0$. The PEKS can be instantiated by Ref. [38] or the improved scheme [39]. As in the delegation procedure, ID_{White} and $\mathcal{IBE}$ denote standard ID-based public key and encryption, respectively. The pre-shared secret key ν is assumed to exist between the delegator Dr. White and storage server SS in the same organization/management domain or can be easily established otherwise. The keyword KW can be the delegator's identity, date/time the encryption is created, etc., for retrieving the intended patient data. The choice of keywords must obey an agreed-upon syntax so that the delegatee will be able to specify proper keywords for searching. The single keyword PEKS shown above can be extended to facilitate multiple-keyword search [40].

When the delegatee Dr. Black needs to access the intended patient data after being delegated, Dr. Black first authenticates with the storage server SS to prove appropriate access permissions, leveraging the proxy signatures obtained from the proxy signer. The storage server SS essentially performs the same proxy signature verifications as the delegatee, to assure the authenticity and permissions of both the proxy signer and the delegatee. We have omitted these straightforward authentication steps in the following illustration to avoid repeating, and assume the establishment of a shared secret key $\pi = \overline{a}\overline{b}P_0$ during the above authentication, analogous to the establishment of ζ. After successful authentication for access (with basic revocation implicitly executed), SS needs to check if Dr. Black is revoked on demand by Dr. White. If not revoked, Dr. Black can obtain the encrypted patient data from SS as follows [37]:

1. $Black \rightarrow SS$: IDT_R, $TD_R(KW)$, t_9, $\mathcal{HMAC}_\pi(PKT_R \parallel TD_R \parallel t_9)$,
2. $SS \rightarrow Black$: $\mathcal{HIBE}_{PKT_R}(PatientData)$, t_{10}, $\mathcal{HMAC}_\pi(\mathcal{HIBE} \parallel t_{10})$,

where $TD_R(kw) = \tau H_2(KW)$ is the trapdoor computed by the delegatee Dr. Black for searching KW. The storage server SS searches over the encrypted patient data designated for Dr. Black and returns the results $\mathcal{HIBE}_{PKT_R}(PatientData)$ to Dr. Black.

27.4.9. Revoking Access Rights

Intuitively, after the delegator Dr. White delegates access rights to the delegatee Dr. Black, Dr. White will need to revoke such rights at a later time for various reasons, including completion of a task, availability issues or role change of Dr. Black, abortion of collaboration with Dr. Black due to Dr. White's dissatisfaction, etc. In this section, we discuss issues around revocation in health information sharing.

Construction of Warrant. In most cases, it would suffice that the delegator only specifies the expiry date of the delegated rights, indicating the expected termination time of a cooperative task. The expiry date can be easily incorporated into the warrant W. We thus do not further pursue solutions to common-case revocation which will in fact be part of the solutions to on-demand revocation. In our EHR system, two exceptions may occur which cannot be dealt with by simply using the expiry date. First of all, the role-based delegation procedure delegates access rights to a role instead of a delegatee. Whenever the delegatee becomes unavailable (e.g., due to a busy schedule), the policy server PS needs to reselect a delegatee in the same role from possible candidates. The main advantage of this mechanism lies in the transparency at the delegator's side, in that the delegator need not be involved in a new delegation procedure caused by the internal changes of the cooperating organization. This mechanism also greatly reduces communication overhead in the system incurred by cross-domain authentication, particularly when the internal changes take place frequently. However, it renders revocation more difficult, in that the delegator cannot directly revoke a role (and hence any entity in that role). The delegator can only revoke the appointed proxy signer, according to the original design of the warrant [17] which includes only the proxy signer's public key and related information. In the presentation of the delegation procedure, we have implied our solution by incorporating the role PKT_R in the warrant. In the design of warrant shown below, we will elaborate on this issue. The second exception would be the aforementioned abortion of cooperation due to the delegator's dissatisfaction, or situations alike, that require immediate revocation of delegated rights regardlessly. Expiry dates alone cannot cope with such unpredictable or on-demand revocation requests.

The warrant W plays a central role in our revocation procedure, which is constructed as [37]:

$$\begin{aligned}
PS:\ & EXP_0,\ PKT_{PS},\ \mathcal{HIBE}_{PKT_{PS}}(\Delta_0),\ MISC_0;\\
R_1:\ & Rheumatologist,\ EXP_1/N_1,\ PKT_{R_1},\\
& \mathcal{HIBE}_{PKT_{R_1}}(\Delta_1),\ MISC_1;\\
R_2:\ & Cardiologist,\ EXP_2/N_2,\ PKT_{R_2},\\
& \mathcal{HIBE}_{PKT_{R_2}}(\Delta_2),\ MISC_2;\\
R_3:\ & Eye\_Specialist,\ EXP_3/N_3,\ PKT_{R_3},\\
& \mathcal{HIBE}_{PKT_{R_3}}(\Delta_3),\ MISC_3;\\
\vdots\ & \vdots\\
R_6:\ & Pediatrician,\ EXP_6/N_6,\ PKT_{R_6},\\
& \mathcal{HIBE}_{PKT_{R_6}}(\Delta_6),\ MISC_6;\\
\vdots\ & \vdots
\end{aligned}$$

where EXP_*, N_*, PKT_{R_*}, and $MISC_*$ denote the expiry date/time of the proxy signing rights or the delegated access rights, the number of times a delegatee can access the encrypted patient data, the public key associated with the descriptive string of a role R_*, and some miscellaneous restrictions, respectively. The encryption of a public key Δ_*, $\mathcal{HIBE}_{PKT_{PS}}(\Delta_0)$, issued by the delegator will be used for the second exception mentioned above. These encryptions intended for roles in the warrant will be delivered by the proxy signer PS to corresponding delegatees in those roles, during the Step 2 of the delegation procedure (not actually shown).

On-Demand Revocation. The warrant W specifies the proxy signer PS and all possible roles R_* the delegator is likely to cooperate with. Note that it is possible to add in additional roles, which requires reissuance of warrant by the delegator (i.e., generating a new signature $\mathcal{HIDS}_{\psi_{White}, s_{White}}$ on the warrant) [37]. One key point of using role-based delegation here is that though it would be cumbersome to list all possible names of delegatees, it is possible to exhaust all possible roles in an organization, or at least those the delegator will most frequently encounter, so that the reissuance of warrant occurs infrequently. In addition, the types of roles in an organization are expected to remain unchanged while the employees in those roles may undergo frequent and drastic changes. This design rationale also lends itself to the solution of the first exception in revocation. Since each role and associated restrictions are specified in the warrant, which are absent in the original warrant design according to Ref. [17], the delegator now has fine-grained control over the revocation of a role besides merely relying on the proxy signer. Without the specification of role restrictions, revocation of a role will have to leverage the proxy signer as the "messenger." Even if a delegated role, say, R_3 expires (or violates other restrictions), the warrant need not be modified or reissued and can be continuously used until all entries have expired. Note that N_*, which restricts the number of times the delegatee (in a role) can access the encrypted patient data, is included for fine-grained revocation compared with the general expiry date/time. It manifests another merit of role-based delegation/revocation. When the delegatee Dr. Black as the rheumatologist first access the intended patient data, the storage server SS initiates a counter $n = N_*$ and decreases it each time the corresponding role PKT_{R_*} accesses the patient data. Suppose Dr. Black is later replaced by Dr. Brown to take over the role as a rheumatologist, n will continue to count down instead of being refreshed to a new value since Dr. Black and Dr. Brown are under the same role-based public key PKT_{R_*}.

To deal with the second exception case, a more powerful revocation mechanism will be needed since this exception case has the most stringent requirement that the delegator must be capable of revoking any delegated role at any time [37]. An intuitive approach would be based on the idea of certificate revocation list (CRL), where the delegator updates the list of revoked public keys PKT_{R_*} and distributes it to the storage server. This approach is appropriate if we assume the storage server cannot be compromised. In reality, the storage server can be impersonated by malicious attackers or be corrupted by the delegatee (i.e., collusion attack), so that the storage server will allow the revoked delegatee with PKT_{R_*} to continue accessing the patient data. This security breach exists due to the lack of control over the revocation at the storage server. A suitable solution would use more advanced technique, the dynamic accumulator [48]. The idea behind is to consider the delegator to be running a dynamic group of delegated roles, each of which has a public key Δ_* assigned by the delegator. The delegator publishes and updates an archive ARC on the storage server recording past and current accumulated values. The delegator also publishes public parameters $\Theta \in_R G_1$ and $\Theta_{pub} = s\Theta$ on the storage server, with $s \in_R Z_q^*$ the secret information of the delegator. The accumulated value is updated whenever the delegator has (a) a new role to delegate which is not in the constructed warrant (i.e., the delegator needs to reissue a warrant incorporating this new role), (b) an existing role in the warrant to delegate (i.e., the warrant can contain *schedules of future delegations* if the restrictions to be applied can be predetermined), or (c) a delegated role to revoke on demand. Cases (a) and (b) represent the joining to the delegator's dynamic group, which is essential for the later on-demand revocation as in Case (c). Before the delegatee authenticates with the storage server to access the patient data, the delegatee needs to update a witness ϖ_* associated with the assigned public key Δ_*, such that the witness cannot be successfully updated if this delegatee is revoked. Note that the delegatee can skip the witness update if the accumulated value remains unchanged (i.e., none of Cases (a), (b), (c) happens).

For example, the delegator Dr. White has distributed the public key Δ_6 to the delegatee Dr. Black as Pediatrician R_6 and published necessary information on the storage server SS, as described above. Assume that ARC currently comprises k accumulated values up to V_k. The on-demand revocation procedure based on dynamic accumulators we just presented is illustrated as follows, which occurs in between the proxy signature verification (including inspection on the warrant for EXP_*/N_* based common-case revocation) and the PEKS-encrypted data retrieval mentioned previously [37]:

1. $Black \rightarrow SS$: $\Phi = r^{-1}\varpi_k$, $\Psi = r(\Delta_6\Theta + \Theta_{pub})$, t_{11}, $\mathcal{HMAC}_\pi(\Phi \parallel \Psi \parallel t_{11})$;
2. $SS \rightarrow Black$: $\hat{m}$, t_{12}, $\mathcal{HMAC}_\pi(\hat{m} \parallel t_{12})$,

where $r \in_R Z_q^*$ is the random secret selected by Dr. Black, π is the shared secret key established during the proxy signature verification, and $\varpi_k = Update(\iota, \varpi_\iota)$ with $Update$ the witness update algorithm in Ref. [49] and $\iota < k$ the last update instance. In Step 2, the storage server SS checks the revocation status of the delegatee Dr. Black by verifying the equality $e(\Phi, \Psi) = e(V_k, \Theta)$. If it holds, the delegatee Dr. Black is not revoked, otherwise Dr. Black is revoked. The most updated accumulated value V_k is calculated by the delegator Dr. White and published on the storage server SS as follows: (1) $V_k = (1/s + \Delta_6)V_{k-1}$ if Dr. Black was revoked in the $k-1st$ instance, or (2) $V_k = (s + \Delta_*)V_{k-1}$ if there is joining to the dynamic group (i.e., either Case (a) or (b) happens) and no one was revoked in the $k-1st$ instance, or (3) V_k remains unchanged if there is no revocation or joining. The storage server then sends a message $\hat{m}$ to Dr. Black indicating the revocation status of Dr. Black. Refer to Ref. [49] for the $Update$ algorithm, the initialization of the accumulated value V_0, and more details on the dynamic accumulator operation.

Discussion. One may argue that a sophisticated attacker can still subvert the execution of the storage server and deviate the dynamic-accumulator-based revocation. Note that in general, such tampering attacks cannot be resolved by any cryptographic scheme, which only functions in the face of unsophisticated attackers who are assumed unable to tamper with/modify the software or hardware of a compromised entity. The same assumption exists in many application scenarios with applied cryptography [50]. In the aforementioned intuitive approach, the list of revoked public keys PKT_{R_*} must be signed by the delegator to guarantee integrity, since otherwise even an unsophisticated attacker can easily alter a public key, resulting in a supposed revocation unattainable. Whereas in our dynamic-accumulator-based approach, altering the information stored at the storage server only renders a supposed successful verification to fail, causing the entity under verification to be revoked. Apparently, if the attacker's goal is to bypass the revocation mechanism and enable a revoked delegatee to continue accessing patient data, our approach will thwart such attacks. More importantly, the dynamic-accumulator-based approach allows our system to extend to incorporate anonymous settings. Although not within the scope of this chapter, it is sometimes desirable to hide the real identity of the delegator and delegatee, and their cooperative relationship, during communications. These pieces of information can be sensitive data in some applications, especially when the cooperative relationship must be kept confidential in the health-care industry. A common technique to achieve such privacy is through the use of pseudonyms in place of the public key PKT_{R_*} which reveals identity information. Then revocation based on updating and distributing the list of pseudonyms (in the anonymous settings instead of PKT_{R_*}) will be highly difficult since pseudonyms are changed frequently to preserve privacy. On the other hand, our dynamic-accumulator-based approach can combine with the pseudonym technique to achieve both revocation and privacy protection for the communicating parties. Furthermore, adopting our approach provides a countermeasure to the collusion attack where the delegator colludes with the storage server to revoke any delegatee of choice. This collusion attack cannot be resisted by the intuitive approach, while can be resisted by the dynamic-accumulator-based approach [49]. Last but not least, the dynamic-accumulator-based revocation mechanism also outperforms the group-signature-based counterpart in terms of both security and efficiency. The detailed comparisons can be found in Ref. [48].

We have mentioned that if revocation is used, the outsourcing of PEKS-encrypted data enabling distributed storage would be intractable, in that the outside storage servers cannot be trusted to execute the revocation mechanism. If the storage server is totally public which is outside the EHR system, it is impossible to apply any technique for such server (due to the difficulty in authentication) to exercise revocation. Therefore, the option of outsourcing to a totally public server is feasible only when revocation is not needed. In systems where distributed data storage/retrieval is attractive, we can leverage the storage server at each delegatee's organization, or a public server designated for patient data storage within the EHR system (i.e., the public storage server shown in Figure 27-1), to host the PEKS storage and retrieval. Since the trust relationship with such cross-domain storage servers can be established via HIB-PKI, we can use delegation once again to allow such servers to act on behalf of the delegator, the design details of which are beyond the scope of this chapter. Please refer to Refs. [31, 37] for more details on protecting patient privacy, controlling access to patients' health records, delegation, fine-grained access control, and on-demand revocation.

27.5. SECURITY ANALYSIS

In this section, we show that the proposed HIPS system satisfies the predefined security requirements [31, 37].

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 Ref. [14] 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. The confidentiality of the patient data shared between the delegator and delegatee to facilitate cooperation is assured by the PEKS primitive, which essentially protects patients' health data privacy. Furthermore, secret information contained in message exchanges remains confidential by using encryption schemes (i.e., $\mathcal{HIBE}$, $\mathcal{IBE}$).

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 HIPS 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. In addition, fine-grained access control has been ensured by the *minimum-privilege delegation* requirement. The goal of the delegatee is to ultimately access the PEKS-encrypted patient data with proper assigned role. However, in order to prevent other entities in a same role who are not delegated to access the data, the proxy signature-based basic access control has to be in place. By performing proxy signature verification, the verifier can be ascertain of an entity's delegation status.

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.

Minimum-Privilege Delegation: This requirement is specially treated with the fine-grained access control. The role-based delegation is too general to restrict the delegated privilege to be precisely minimum necessary, which is the reason to use the PEKS-based access control. Through encrypting the exact patient data to be accessed by the delegatee, the delegator actively and effectively restricts the privilege of the delegatee to only those data intended or PEKS-encrypted for later retrieval.

Adaptability: By using the role-centered approach for both delegation and revocation, instead of more specific entity-based approach, the changing credentials or availability of a delegatee is fully dealt with by the policy server of the delegatee's organization, without intervention of the delegator or interruption of any delegation and revocation procedures, which are thus transparent to the delegator.

On-Demand Revocation: The advanced revocation mechanism based on dynamic accumulators is dedicated to handling on-demand revocation. Through the update of the accumulated values performed by the delegator and maintained at the storage server, the delegatee with a revoked public key will be automatically restrained from further access. This technique is also very efficient in that the storage cost incurred by the accumulated values is independent of the number of revoked entities or the total number of entities in the delegator's dynamic group.

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 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.

Data Integrity: Since the PHI and MHI are encrypted, no one except the patient him/her self 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 distributed to *any other* entity. The protocol message integrity is ensured by including HMAC or digital signatures in the message exchanges. Moreover, the integrity of both the stored patient data and the exchanged messages during interactions is guaranteed by either signature schemes (i.e., $\mathcal{HIDS}$, $\mathcal{IBS}$), or message authentication code (i.e., $\mathcal{HMAC}$).

Authenticity: This objective is achieved leveraging cross-domain and in-domain authentication based on hierarchical ID-based signature $\mathcal{HIDS}$ and standard ID-based signature $\mathcal{IBS}$, respectively.

Another requirement on the delegation in distributed EHR system is the onward delegation [15], where the appointed proxy signer is able to further appoint other proxy signers in order to complete a task, thereby forming a delegation chain. Since our EHR system is concerned with the sharing of sensitive patient data, it is inappropriate to enable delegation chain which would be in violation of our fine-grained access control rationale. Intuitively, fine-grained access control is developed due to the delegator's insufficient control over the delegatee's access to patient data. If the delegation chain is allowed by the delegator, the proxy signer can further delegate an entity without the delegator's awareness. In other words, the delegator voluntarily surrenders some of his/her control power. In some other application scenarios, e.g., cooperative research for statistical studies, where patient data have been preprocessed to remove sensitive information, the delegation chain can be adopted to support onward delegation and alleviate the burden on the delegator. In this case, our proxy signature-based delegation can be easily adapted to provide onward delegation mechanism [15].

27.6. CONCLUSION AND FUTURE WORK

In this chapter, we present a secure EHR system to protect patient privacy, and at the same time, to enable emergency health care and support patient data sharing across cooperative organizations. The system is demonstrated to be resilient to various attacks, fulfill the desired functionalities, and satisfy the security requirements.

There are many open issues that constitute our future work. The first two issues are related to the security aspects of HIPS. The remaining issues are pertinent to the implementation, evaluation, and deployability of HIPS.

1. Attacks and Threats: Besides social engineering, there are potential attacks to HIPS that will cause devastating consequences. Such attacks comprise collusion, traffic analysis, timing analysis, and denial-of-service (DoS), which we have identified in the design of HIPS. Consequently, additional mechanisms such as alerting functions of the P-device, anonymous communication substrate, randomized scheduling for PHI uploads, decentralized storage (to S-servers), and authentication (to A-servers) have been incorporated into HIPS rendering the attacks extremely difficult. Nonetheless, there will be unknown adversarial behavior that can lead to effective attacks to HIPS, especially when computing and communication technologies are evolving fast. As a result, new threats must be identified and defined in the evaluation phase and only when HIPS is road-tested on trial deployment or even real systems.
2. Social Engineering: This is a special and powerful type of threats to any security systems with human intervention. Human is arguably the weakest link in secure communication systems. This weakness is often exploited by attackers to break into the system where the security architecture is well-designed and hard to attack by other means. Possible forms of social engineering include tailgating, spidering, etc. [46]. The effective social engineering threats to HIPS, an example being the (attackers') exploitation in PHI access control and dissemination in the context of social networks, are also possible to be quantitatively measured. For instance, the risk of victims being identified or linked to certain activities can be measured in terms of *anonymity*; the probability that an adversary's hypothesis is true, given the various information gained by the adversary over social networks [51]. This probability is calculated by Bayesian inference as $Pr(h_j|E) = (Pr(h_j)\,Pr(E|h_j)/\sum_{k=1}^{N} Pr(h_k)\,Pr(E|h_k))$, where $Pr(h_*)$ is the a priori probability of hypothesis h_* being true, and $Pr(E|h_*)$ is the conditional probability of the event E being true given that h_* is true. The number of hypotheses N indicates possible sources of information from the social network for concluding the hypotheses. The Bayesian inference technique can be applied to HIPS to quantify the impact of social engineering threats, given the entity interaction patterns in our application scenario. The corresponding countermeasures will be developed subsequently.
3. Development of Metrics: The evaluation of system performance currently takes place in the form of analytical reasoning. More

quantitative and practical evaluations need to be carried out, including simulation and trial deployment of HIPS on real systems, which cannot be easily accomplished lacking the right tool, the metrics for quantitative measurement. Our first task will be to explore meaningful and measurable quantities that can serve as metrics for evaluating the system performance. For example, the amount of anonymity guaranteed by HIPS can be measured leveraging Bayesian inference as mentioned above. Another example would be the anonymization of SHI to remove identification data of the patient, for the collaborative clinical research in HIPS. Meaningful metrics can be re-identification risk and information loss [52], measured by re-identification probability and discernability, respectively. The re-identification probability is defined by $\vartheta_{max} = (1/min_j(F_j))$, where $F_j = \sum_{i \in U} I(X_{Z,i} = j)$ $(j = 1, \ldots, J)$ is the size of an equivalence class in the identification database Z, the records in which take on J different values and have a one-to-one mapping to the individuals in U, and $X_{Z,i}$ denotes the value of record i in Z. The discernability metric [53] is defined as: $C(g,k) = \sum_{\forall E, |E| \geq k} |E|^2 + \sum_{\forall E, |E| < k} |D||E|$, where E refers to the equivalence classes of tuples (of size k) in the data set D induced by the anonymization g and $||$ denotes the size or cardinality. We will investigate the applicability of these metrics to HIPS, and more importantly, develop other key metrics capturing measurable performance pertinent to security, privacy, and trustworthiness of HIPS.

4. System Deployment and Evaluation: At the current stage, system evaluation is performed by means of analytical reasoning, which demonstrates that the state-of-the-art design of HIPS fulfills the predefined security requirements. Our future work consists of simulation using developed metrics, trial deployment possibly cooperating with The University of Florida Healthcare Center (also known as Shands Hospital), to study the tradeoffs among communication, computation, and storage, ensure no vulnerability or inefficiency is introduced as a byproduct during HIPS design, and test the interoperability and performance of the system.

5. Human Operations and Autonomic System: Human and technology are interwoven. As indicated in the social engineering threats, even the most secure system could be easily hacked by exploiting human factors. It is thus attractive to reduce human intervention, a possible solution of which is to use autonomic communications system [54, 55] as a substitute for human decisions and operations. Our health information sharing functionality can be implemented autonomically, resulting in greatly simplified operations from end users' perspective. A proof-of-concept implementation of autonomic information sharing between two health-care providers is currently being developed as shown in Figure 27-3. The autonomic elements designated for information sharing in Figure 27-3 are at a higher level of the autonomic communications system above individual computing elements level. In general, an autonomic element is constituted by a functional unit which performs basic operational functions and a management unit which monitors and controls the operation/configuration of the functional unit. These autonomic elements are expected to operate in a very dynamic environment, with only fixed policies (i.e., privacy and security policies in Figure 27-3) and functions consisting of authentication, delegation, access control, and revocation. These functions which will be realized through decomposing the information sharing element to four lower-level elements performing the corresponding four functions.

For our ongoing research, smart context-aware autonomic system with intrusion detection functionalities would be an ideal candidate for reacting to unforeseeable attacks and informing users in a timely fashion. On the other hand, human operations are indispensable in critical-decision making

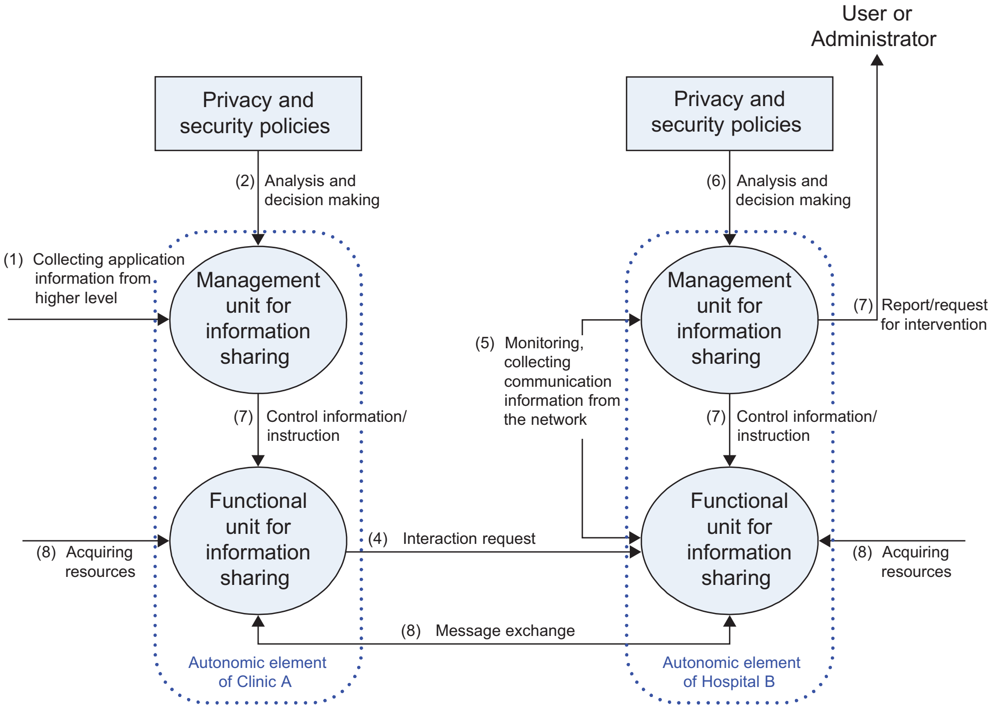
User or Administrator
Privacy and security policies
(2) Analysis and decision making
Privacy and security policies
(6) Analysis and decision making
(1) Collecting application information from higher level
Management unit for information sharing
Management unit for information sharing
(7) Report/request for intervention
(5) Monitoring, collecting communication information from the network
(7) Control information/ instruction
(7) Control information/ instruction
Functional unit for information sharing
(4) Interaction request
Functional unit for information sharing
(8) Acquiring resources
(8) Acquiring resources
(8) Message exchange
Autonomic element of Clinic A
Autonomic element of Hospital B

FIGURE 27-3 Logic diagram of our autonomic health information sharing framework.

and stringent control of sensitive PHI and SHI. Therefore, the design of HIPS should not introduce inconvenience or complexity to human users in performing these operations. Whether autonomic communications system is appropriate for HIPS, and how to guarantee security and privacy of HIPS while still rendering it usable for human users, remain open questions for us to seek answers.

As a final remark, many of the open issues we have brought forward are present in almost all security systems. Success in tackling these issues will enlighten research on similar issues in other application systems and contribute to deeper understanding in security and trustworthiness.

ACKNOWLEDGMENTS

This work was partially supported by the U.S. National Science Foundation under grants CNS-0916391 and CNS-0716450, the National Natural Science Foundation of China under Grant 61003300, and the China 111 Project under Grant B08038. The work of X. Zhu was also partially supported by the Fundamental Research Funds for the Central Universities under grant JY10000901021.

EXERCISE PROBLEMS

1. What is EHR? Give some examples of EHRs. What are the advantages of EHRs over paper-based medical records?
2. What are the three requirements of privacy?
3. In the hierarchical ID-based cryptosystem used by the authentication of our e-health-care system, can a parent learn the private key of a child?
4. Name two conflicting requirements for a secure and functional e-health-care system and how to solve the conflict.
5. What technique is used to protect patient privacy against the storage server? How does this technique achieve this goal?

6. In protecting patient privacy and providing access control, we used searchable symmetric encryption and public-key encryption with keyword search (PEKS) both for securely searching over encrypted data. Describe the key differences between these two schemes in terms of their different constructions and uses.
7. How to guarantee location privacy when patients use their mobile devices to access the e-health-care system (suppose the locations of the device can be tracked all the time such as cell phones)?
8. When is dynamic revocation desirable and how to achieve it?
9. Can you think of a meaningful attack to the proposed e-health-care system that was not mentioned in the chapter and the countermeasure?
10. What are the challenges of implementing health information sharing in an autonomic communications environment?

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 (accessed 08.18.08).

[3] M.C. Rash, Privacy concerns hinder electronic medical records, Bus. J. Greater Triad Area. April 2005.

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

[5] R. Pear, Warnings Over Privacy of U.S. Health Network, New York Times, Feb. 2007.

[6] U. Sax, I. Kohane, K.D. Mandl, Wireless technology infrastructures for authentication of patients: PKI that rings, J. Am. Med. Inf. Assoc. 12 (3) (2005) 263–268.

[7] M.C. Mont, P. Bramhall, 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), Prague, Czech Republic, 2003.

[8] T. Denning, K. Fu, 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), San Jose, California, July 2008.

[9] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, et al., Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses, in: Proc. IEEE Symposium on Security and Privacy, Oakland, California, May 2008.

[10] R.S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, Role-based access control models, Computer, 29 (2) (1996) 38–47.

[11] L. Zhang, G. J. Ahn, B. T. Chu, A role-based delegation framework for healthcare information systems, in: SACMAT, Monterey, California, 2002, pp. 125–134.

[12] L. Zhang, G. J. Ahn, B. T. Chu, A rule-based framework for role-based delegation and revocation, ACM Trans. Inf. Syst. Secur. 6 (3) (2003) 404–441.

[13] W.-B. Lee, C.-D. Lee, A cryptographic key management solution for HIPAA privacy/security regulations, IEEE Trans. Inf. Technol. Biomed. 12 (1) (Jan 2008) 34–41.

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

[15] M. Katzarova, A. Simpson, Delegation in a Distributed Healthcare Context: A Survey of Current Approaches, in: S. K. Katsikas et al. (Eds.), ISC 2006, LNCS, vol. 4176, Springer-Verlag, Palermo, Italy, 2006.

[16] V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, et al., X.509 Proxy Certificates for Dynamic Delegation, in: third Annual PKI R&D Workshop, Gaithersburg, Maryland, 2004.

[17] A. Boldyreva, A. Palacio, B. Warinschi, Secure Proxy Signature Schemes for Delegation of Signing Rights, Cryptology ePrint Archive, Report 2003/096, available at http://eprint.iacr.org/2003/096.pdf, 2003.

[18] M. Gasser, E. McDermott, An architecture for practical delegation in a distributed system, in: Proc. IEEE Symposium on Research in Security and Privacy, Oakland, California May 1990, pp. 20–30.

[19] R. Housley, W. Polk, W. Ford, D. Solo, Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, RFC 3280, April 2002.

[20] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, A security architecture for computational grids, in: ACM Conference on Computer and Communications Security (CCS), San Francisco, California, 1998, pp. 83–92, 1998.

[21] G. Navarro, B. Sadhigi-Firozabadi, E. Rissanen, J. Borrell, Constrained delegation in XML-based access control and digital rights management standards, December 2003.

[22] L. Seitz, E. Rissanen, T. Sandholm, B. S. Firozibadi, O. Mulmo, Policy administration control and delegation using XACML and delegent, in: 6th IEEE/ACM International Workshop on Grid Computing, Seattle, Washington, November 2005.

[23] J. Wang, D. D. Vecchio, M. Humphrey, Extending the security assertion markup language to support delegation for web services and grid services, in: IEEE International Conference on Web Services (ICWS'05), Orlando, Florida, July 2005.

[24] Oasis eXtensible Access Control Markup Language Committee, XACML V2.0, http:www.oasis-open.org/committees/.

[25] G. Ateniese, R. Curtmola, B. de Medeiros, D. Davis, Medical information privacy assurance: cryptographic and system aspects, in: 3rd Conference on Security in Communication Networks (SCN'02), Amalfi, Italy, September 2002.

[26] A.J. McMurry, C.A. Gilbert, B.Y. Reis, H.C. Chueh, I.S. Kohane, K.D. Mandl, A self-scaling, distributed information architecture for public health, research, and clinical care, J. Am. Med. Inform. Assoc., 14 (4) (2007) 527–533.

[27] D.J. Power, E.A. Politou, M.A. Slaymaker, A.C. Simpson, Towards secure grid-enabled healthcare, Software: Prac. Exp., 35(9) (2005) 857–871.

[28] J. Sun, X. Zhu, Y. Fang, Privacy and emergency response in e-healthcare leveraging wireless body sensor networks, IEEE Wireless Commun. 17 (1) (2010) 66–73.

[29] C.W. Burt, E. Hing amd D. Woodwell, Electronic medical record use by office-based physicians: United states, 2005, National Center for Health Statistics, http:www.cdc.gov/nchs/products/pubs/pubd/hestats/electronic/electronic.htm, 2005.

[30] CDC's National Center for Health Statistics, More physicians using electrical medical records, http://www.cdc.gov/od/oc/media/pressrel/a060721.htm, 2006 (accessed 08.18.08).

[31] J. Sun, X. Zhu, C. Zhang, Y. Fang, HCPP: Cryptography Based Secure EHR System for Patient Privacy and Emergency Healthcare, IEEE International Conference on Distributed Computing Systems (ICDCS'11), June 2011.

[32] J. Sun, X. Zhu, Y. Fang, Preserving Privacy in Emergency Response Based on Wireless Body Sensor Networks, in: Proceedings of the IEEE Global Communications Conference, December 2010.

[33] C.-H. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J.M. McCune, A. Studer, et al., GAnGS: Gather, Suthenticate 'n Group Securely, in: Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom), San Francisco, California, September 2008.

[34] C.-H. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, et al., Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication, in: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, May 2005.

[35] L. Zhong, M. Sinclair, 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), Cambridge, Massachusetts, 2006.

[36] 45 C.F.R. part 160 — general administrative requirements, 160.103: definitions, Department of Health and Human Services, vol. 1, October 2002.

[37] J. Sun, Y. Fang, Cross-domain data sharing in distributed electronic-health-record system, IEEE Trans. Parallel. Distrib. Syst. 21 (6) (2010) 754–764.

[38] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano, Public Key Encryption with Keyword Search, in: EUROCRYPT 2004, LNCS, vol. 3027, Springer, Interlaken, Switzerland, 2004.

[39] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, et al., Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions, in: V. Shoup (Eds.), Crypto 2005, LNCS, vol. 3621, Springer, Santa Barbara, California, 2005.

[40] J. Baek, R. Safiavi-Naini, 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 (accessed 08.18.08).

[41] U.S. Department of Health & Human Services Website, Health information technology, http://www.hhs.gov/healthit/ (accessed 08.18.08).

[42] H.W. Lim, K. G. Paterson, Identity-based cryptography for grid security, in: H. Stockinger, R. Buyya, R. Perrott (Eds.), in: Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005), IEEE Computer Society Press, Melbourne, Australia.

[43] C. Gentry, A. Silverberg, Hierarchical id-based cryptography, in: Proc. ASIACRYPT, Queenstown, New Zealand, December 2002, pp. 548–556.

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

[45] J. Sun, C. Zhang, Y. Fang, A security architecture achieving anonymity and traceability in wireless

mesh networks, IEEE Conf. on Computer Communications (INFOCOM), April 2008, pp. 1687–1695.

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

[47] F. Hess, Efficient Identity-Based Signature Schemes Based on Pairings, SAC 2002, LNCS, vol. 2595, Springer-Verlag, Madrid, Spain, 2002, pp. 310–324.

[48] J. Camenisch, A. Lysyanskaya, Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials, CRYPTO 2002, vol. 2442, Springer-Verlag, Santa Barbara, California, 2002, pp. 61–76.

[49] L. Nguyen, R. Safavi-Naini., Dynamic k-times anonymous authentication, in: Applied Cryptography and Network Security Conference, vol. 3531, New York City, New York, 2005, pp. 318–333.

[50] T. Ristenpart, G. Maganis, A. Krishnamurthy, 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, San Jose, California, July 2008, pp. 275–290.

[51] C. Diaz, C. Troncoso, A. Serjantov, On the impact of social network profiling on anonymity, in: N. Borisov, I. Goldberg (Eds.), PETS 2008, LNCS, vol. 5134, Springer-Verlag Berlin, Heidelberg, 2008.

[52] K.E. Emam, F. K. Dankar, Protecting privacy using k-anonymity, J. Am. Med. Inf. Assoc. 15 (5) (2008) 627–637.

[53] R. Bayardo, R. Agrawal, Data privacy through optimal k-anonymization, in: Proceedings of the 21st International Conference on Data Engineering, Tokyo, Japan, 2005, pp. 217–228.

[54] S. Dobson et al., A survey of autonomic communications, ACM Trans. Auton. Adapt. Syst. 1 (2) (2006) 223–259.

[55] D.M. Chess, C. C. Palmer, S. R. White, Security in an autonomic computing environment, IBM Syst. J. 42 (1) (2003) 107–118.