2012 32nd IEEE International Conference on Distributed Computing Systems

PAAS: A Privacy-Preserving Attribute-based Authentication System for eHealth Networks

Linke Guo*, Chi Zhang†, Jinyuan Sun‡ and Yuguang Fang*
*Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
†School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China
‡Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA
Email: blackglk@ufl.edu, chizhang@ustc.edu.cn, jysun@eecs.utk.edu, fang@ece.ufl.edu

***Abstract*—Recently, eHealth systems have replaced paper based medical system due to its prominent features of convenience and accuracy. Also, since the medical data can be stored on any kind of digital devices, people can easily obtain medical services at any time and any place. However, privacy concern over patient medical data draws an increasing attention. In the current eHealth networks, patients are assigned multiple attributes which directly reflect their symptoms, undergoing treatments, etc. Those life-threatened attributes need to be verified by an authorized medical facilities, such as hospitals and clinics. When there is a need for medical services, patients have to be authenticated by showing their identities and the corresponding attributes in order to take appropriate healthcare actions. However, directly disclosing those attributes for verification may expose real identities. Therefore, existing eHealth systems fail to preserve patients' private attribute information while maintaining original functionalities of medical services. To solve this dilemma, we propose a framework called PAAS which leverages users' verifiable attributes to authenticate users in eHealth systems while preserving their privacy issues. In our system, instead of letting centralized infrastructures take care of authentication, our scheme only involves two end users. We also offer authentication strategies with progressive privacy requirements among patients or between patients and physicians. Based on the security and efficiency analysis, we show our framework is better than existing eHealth systems in terms of privacy preservation and practicality.**

***Index Terms*—Authentication, non-interactive zero-knowledge proof, non-interactive witness-indistinguishable, homomorphic encryption**

I. Introduction

The widely deployed electronic health (eHealth) system has changed people's daily life other than traditional paper based system for its extraordinary advantages, such as more efficiency, high accuracy and broader availability. However, privacy concern is arguably the major barrier that hinders the development of electronic health record (EHR) stored in a public storage with direct connection to a network. For most eHealth systems, physicians periodically upload their observations and diagnosis to one particular storage, where protected health information (PHI) is seamlessly bound to the real identity of a specific patient. When physicians are authorized, they can easily obtain both the real identity and designated diseases of a particular patient, which apparently discloses the patient's privacy. To some extent, patients are reluctant to contact a doctor or a medical facility based on the real identities, instead, they prefer to show a token which can represent their diseases or other attributes rather than exposing real identities, and physicians can treat them using the token only. This perfect solution leads us to separate attributes from identity, which brings several open problems related to the system architecture. First, if the authentication process takes place on a centralized authority, even if the identity is isolated from the corresponding attributes, we still need to disclose certain information regarding the relationship between attributes and identity to the authority for verification, so that the centralized authority can process requests and grant privileges to the designated user. On the other hand, if users directly communicate without the help of a central authority, we can guarantee that the privacy issues related to attributes are well preserved. However, purely relying on the distributed users' attributes cannot fulfill the requirement of verifiability of the isolated attributes. In a word, existing eHealth systems lack the ability to satisfy the requirements of preserving the privacy and the verifiability of the corresponding attributes simultaneously. As a result, patients face those security breaches and authentic verification problems when they share the same situation and want to talk with each other via cyber-space. Furthermore, those kinds of concerns become the major barrier that impedes patients from easily communicating [1]. Thus, there is an urgent need for designing a framework where users can authenticate each other using verifiable attributes while keeping their attributes and identities undisclosed.

Related Works: To deal with the potential risks of privacy exposure, several eHealth systems [2]–[4] let patients encrypt their personal health record (PHR) before storing it on the central authority. Although the encrypted PHR prohibits the centralized facility from obtaining the information, it still faces the problem of data verifiability. Since most of those PHRs are vital, physicians cannot accept or utilize the records without an official verification. On the other hand, it seems easy to implement the verification process for the eHealth systems. However, it is obvious that we must directly show the record itself and the corresponding identity to get the PHR verified, all of which bring security breaches to patients. In recent research works on social networks, several possible solutions have been proposed to utilize attributes for authentication without revealing attributes themselves. Similarly, their proposed systems lack the functionality of verifiability. The most relevant work is Li. *et al.* [5] which considers a privacy-preserving personal profile matching scheme for mobile social networks, which implements secure multi-party computation based on polynomial secret sharing. Their scheme is fully distributed, where users share their attributes among a group of valid users using Shamir secret sharing scheme. Also, in [6], they design a secure friend discovery scheme based on secure dot product protocol by using homomorphic encryption. Multiple papers [7]–[9] address the problem of secure private set intersection (PSI), which is related to the highest privacy level in our proposed system. However, none of them considers the verifiability of the private set, which is the major difference compared to our work. More specifically, their schemes deviate from our design goals due to attributes in the eHealth systems are crucial for patients and needed to be verified before taking any further action. For the centralized model, Eagle and Pentland [10]

1063-6927/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDCS.2012.45

IEEE computer society

describe social serendipity to perform matching in mobile social networks, which purely relies on the centralized server. It faces the security breaches of privacy leakage and collusion attacks. In [11], Manweiler. *et al.* propose a novel trust establishment scheme for location-based service, which takes advantage of the location and time information as the key between strangers for recovering potential connections. Their scheme enables people who share the same location and time to reestablish missing connections. However, it has the limitation that the relationship is only relying on the space and time, which is not strongly convincing. Also, since patients in eHealth systems mostly care about their attributes and identity privacy, it is infeasible to render all PHRs to a centralized facility for verification. Thus, none of the above systems satisfies the verifiability and privacy preservation at the same time. The first work of non-interactive zero-knowledge was introduced by Blum *et al.* in [12]. Our scheme employs the non-interactive proof system for bilinear pairing in [13] which has been used for several applications in [14]–[16]. On the other hand, several works regarding attribute-based encryption (ABE) discuss the authentication schemes in [17]–[19]. However, we cannot apply traditional encryption schemes that use shared secret to authenticate strangers. In most attribute-based encryption schemes, the key distribution center is responsible for distributing public/private key pairs based on each individual's attributes and corresponding structure. If they are in the same attribute group, they may mutually authenticate each other. However, in our proposed scenario, patients need to prove to physicians or hospitals that they have a specific disease. Since patients will not share identical attributes with physicians and cannot be verified by using the corresponding secret keys. Furthermore, patients will not expose their sensitive attributes and values for verification, which is also the main factor that the public key cryptosystem will not work.

Our Contributions: In this paper, we design a distributed system for the privacy-preserving authentication among users in eHealth networks. Rather than the conventional approaches which leverage identities to authenticate, our system takes advantage of verifiable attributes to authenticate users without revealing the detail of attributes. The major contribution of this paper is to design a system which simultaneously solves the dilemma: maintaining the privacy and verifiability of attributes of each user (physicians/patients). We offer authentication schemes for four progressive privacy levels for satisfying users' increasing privacy requirements, all of which enable the secure communication between patients and physicians without disclosing identities. Our scheme can prevent common attacks identified in eHealth systems. The experimental results show the feasibility and efficiency of our proposed scheme in detail.

The remainder of this paper is organized as follows. Section II introduces preliminary knowledge of some cryptographic techniques that we use in our system. We describe the system and adversary model in Section III, along with the security objective. The proposed scheme PAAS is presented in detail in Section IV, followed by the performance analysis in Section V. Finally, Section VI concludes the paper.

II. PRELIMINARIES

A. Bilinear Pairing

Bilinear pairing operations are performed on elliptic curves [20]. Let G_1 and G_2 be groups of the same prime order p. Discrete logarithm problem (DLP) is assumed to be hard in both G_1 and G_2. Let P denote a random generator of G_1 and $e : G_1 \times G_1 \rightarrow G_2$ denote a bilinear map constructed by modified Weil or Tate pairing with the following properties:

1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, $\forall P, Q \in G_1$ and $\forall a, b \in Z_p^*$, where Z_p^* denotes the multiplicative group of Z_p, the integers modulo p. In particular, $Z_p^* = \{x \mid 1 \leq x \leq p-1\}$.
2) Non-degenerate: $\exists P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3) Computable: there exists an efficient algorithm to compute $e(P, Q), \forall P, Q \in G_1$.

Bilinear pairing is the basic operation in the identity-based cryptosystem, the non-interactive witness-indistinguishable (NIWI) and zero-knowledge proofs (NIZK), all of which are used as the fundamental techniques in our scheme.

B. NIWI *and* NIZK *proof*

We apply part of the non-interactive proof system in [13], which gives a formal definition for both non-interactive witness-indistinguishable and zero-knowledge proof. We define $\mathcal{R}$ as a computable ternary relation. Given a tuple $(crs, n, w) \in \mathcal{R}$, we call crs as the common reference string, n as the statement that we need to prove and w the witness. Note we also use $\mathcal{L}$ to denote the language consisting of statements in $\mathcal{R}$. Suppose $\mathcal{R}$ consists of three polynomial time algorithms $(\mathcal{K}, \mathcal{P}, \mathcal{V})$, where $\mathcal{K}$ is crs generation algorithm, $\mathcal{P}$ and $\mathcal{V}$ are prover and verifier, respectively. $\mathcal{P}$ takes a tuple (crs, n, w) as input and output a proof π, while $\mathcal{V}(crs, \pi, n)$ will output 1 if the proof is acceptable and 0 if not acceptable. The proof system $(\mathcal{K}, \mathcal{P}, \mathcal{V})$ should satisfy completeness and soundness properties, where completeness denotes if the statement is true, an honest verifier is convinced of this fact by an honest prover, and soundness shows that if the statement is false, and cheating prover can convince the honest verifier that is true with a negligible probability. For NIWI, we require no adversary can distinguish the real crs and simulated crs, while adversaries cannot distinguish which witness the prover uses. For zero-knowledge, we require no verifier obtain additional information other than the fact that the statement is true.

III. SYSTEM MODEL

A. Overview

We first give a brief overview to our proposed system. The main design goal of PAAS is to establish the authentication system in eHealth networks, which leverages the verifiable attributes to authenticate both physicians and patients without compromising each individual's privacy. Apart from schemes that purely rely on the identity, we first define the attribute-based authentication system which simultaneously satisfies the needs of verifiability and privacy-preserving in eHealth networks. Based on different scenarios in the eHealth systems, we show the progressive privacy levels that our system could achieve. As shown in Fig. 1, our system mainly consists of a trust authority (TA) which is responsible for key distribution for users (physicians and patients), a semi-trusted registration center (RC) used to generate and issue credential based on users' attributes in the system. To some extent, TA performs like a government health administration which should be fully trusted, while RC can be hospitals or clinics with certain qualification certified by TA. During the protocol run, RC checks physicians' professionals and issues the corresponding credentials to physicians. Users in different colors in Fig. 1 represent distinct patients in eHealth networks in a distributed manner, and they periodically interact with physicians and obtain credentials or certificates based changed attribute values from RC, such as current symptoms, past medical history, undergoing treatment, etc. For each end user, he/she can either use mobile devices or desktops for the interactions in the networks. Patients can use the pre-assigned pseudonyms to anonymously prove their attributes to communicate with each other and obtain medical services based on the diseases rather than real identities, while physicians can prove their professional skills without showing credentials. We assume users can communicate with each other via wired/wireless links. For simplicity, we also assume that users stay in the transmission range of each other when they use wireless links for communication.

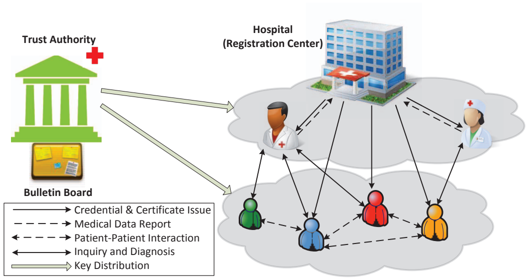

Fig. 1. System Model

B. Security Objective

1) Security Requirements and Assumptions: Our main security objective is to preserve the privacy of each user's identity and attributes. First, we assume that user's attribute set can uniquely identify a particular user, such that we cannot reveal user's attribute in plaintext form during the protocol run. Also, since users use credentials of attributes to authenticate each other, we require the credential of each attribute should be kept undisclosed. Second, our system should be secure under tracing attacks launched by adversaries, which means the information used for verification from the same user in different queries should remain indistinguishable. Otherwise, it is easy for adversary, or even a benign user to trace one particular user. We also make several assumptions which perform like basic building blocks for our proposed system. According to the restrictions and laws, like HIPPA, we forbid physicians and hospitals to distribute PHRs to any unauthorized user. In terms of privacy concerns, patients in our system obtain medical diagnoses based on each attribute provided by patients rather than real identity. Thus, patients can choose pseudonyms to communicate with physicians and/or patients in order to avoid being traced.

2) Privacy Levels: We define four levels of privacy which may potentially satisfy different application scenarios during the interactions in eHealth networks. Note that our four privacy levels model has the progressive property where a higher level of privacy discloses less information but incurs more computation costs, and lower privacy level leaks more details but may be efficient, respectively. However, all of the privacy levels in our framework will provide anonymity and untraceability for users in the system.

Privacy Level 0 (PAAS0): We are defining the most intuitive level of privacy which only requires physicians to show the validity of their professional qualifications on an untrusted third party platform (i.e., social networks). Doctors may want to convince anonymous patients that what they said or suggested is true, but do not want to reveal their credentials or identities in the cyber space. Otherwise, adversaries may impersonate the doctors by using their credentials. Respectively, patients also prefer to use pseudonyms to show their attribute set of a particular disease, where physicians can take turns to verify that the patient indeed faces the illness rather than stealing the remedy. Therefore, PAAS0 requires everyone can verify the validity of the attribute credentials without compromising the privacy of users (physicians or patients).

Privacy Level 1 (PAAS1): Other than verifying the validity of the corresponding attribute credentials, we take a step further to check the value of users' attribute. To some extent, the value of each attribute is a more severe privacy related issue rather than verifiable attributes. For example, to obtain information that Dr. Frank is with the department of internal medicine in Shands hospital at University of Florida will leak less privacy than to know he is a cardiologist in that hospital, where cardiologist is a specific value of an attribute on "*affiliation*". In addition to verifying the validity of attributes, we need the verification of the value of an attribute in the authentication process in the following privacy levels. In PAAS1, users do not care about revealing several kinds of information which is meaningless if it is not associated with real identities. For instance, an AIDS assistant organization provides services but requires a patient's attribute value to satisfy the organization's requirements. Apparently, few patients want to expose the real identities to obtain the services, while PAAS1 satisfies the corresponding conditions and could guarantee the organization can only verify patients' credentials and the value of attributes other than identifying real identities.

Privacy Level 2 (PAAS2): In what follows, we consider the interaction between two patients. Patients may want to share some information concerning their diseases to patients who have the same symptoms, but strictly prohibit other patients to know in detail [21]. Once they learn each other's identical attribute values, they can directly communicate to share certain information. Thus, we need to provide privacy-preserving authentication schemes based on patients' identical attribute values. Rather than the possibility of leaking several "meaningless" information from the patient side in PAAS1, PAAS2 requires patients only reveal the same attribute values to the other users and disclose nothing if two compared attribute values are not identical, in the sense that patients can authenticate each other based on their holding the same verified attribute values while maintaining other attributes undisclosed. When the protocol ends, patient A and B will only mutually learn the intersection set between them: $\mathcal{M}_{AB} = \mathcal{S}_A^* \bigcap \mathcal{S}_B^*$, where $\mathcal{S}_A^* \subseteq \mathcal{S}_A$ denotes the subset of whole attribute set of A. Note that if attributes are Boolean data type, such as the gender, there is no way to prevent the verifier from learning the prover's values of those attributes even if the verification process fails. Thus, we take advantage of several common data types other than Boolean types, i.e., strings and integers.

Privacy Level 3 (PAAS3): For higher security and privacy requirements, PAAS3 requires all of patients' attribute values used for authentication should not be revealed to anyone else. Different from the scenario presented in PAAS2, we require that two patients may only know the cardinality of the intersection set of shared attributes. Taking patient A and B as an example, both A and B only learn the size of the intersection set: $m_{AB} = |\mathcal{S}_A^* \bigcap \mathcal{S}_B^*|$, where $|\mathcal{S}|$ denotes the cardinality of set $\mathcal{S}$. Apparently, we cannot compare each attribute value one by one. Otherwise, it turns out to be the same privacy level as in PAAS2. Instead, we design a verifiable attribute sets comparison protocol for PAAS3 between patients, which only returns the cardinality of intersection set and keeps each of attribute values undisclosed.

C. Adversary Model

We consider various types of adversaries which may launch passive or active attacks to our system. For active attacks, adversaries can launch impersonation attacks to compromise the privacy of both physicians and patients. Also, since our framework can be deployed in a wireless network, an adversary may eavesdrop the communication channels or modify and inject bogus data during the transmissions. On the other hand, it is possible for any kind of user (malicious or benign) to trace back users' real identities, in the sense that attackers may launch active attacks to the identity based on the attribute comparison results. We also need to guarantee the untraceability for any user during the protocol run. Furthermore, we are also concerned with the collusion attack among a group of malicious users or even between RC and malicious users. Since the semi-trusted RC, which is curious but honest, has all credentials it issues, it will largely deteriorate the privacy level if this kind of collusion attacks cannot be thwarted. We will not consider the possibility of sharing secret with others, since this type of active attacks cannot be prevented in most systems.

IV. Proposed Scheme

In this section, we introduce our framework for privacy-preserving authentication in detail. Based on the assumptions and definitions in the previous sections, we first give a formal non-interactive proof system used in this paper and present the progressive privacy levels according to the proposed system architecture.

A. Non-Interactive Proof System

We implement the non-interactive proof system proposed by Groth and Sahai [13], which is an efficient system for bilinear groups. However, their work is based on general cases for bilinear pairing without considering the scenarios in eHealth system. Thus, we give a brief introduction to the system that fits our scenarios.

1) Setup: As in previous section, assume that we have a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. With entry-wise multiplication, we can get the Z_p-modules $M_1 = G_1^3$, in which we define another bilinear map $\hat{e} : M_1 \times M_1 \rightarrow M_2$. Note our system is based on the decision linear assumption introduced by Boneh *et al.* in [22] stating that given three random generators $f, h, g \in G_1$ and f^r, h^s, g^t, it is hard to distinguish the case $t = r + s$ from t random. In the main design of our system, we use the module $M_2 = G_2^6$ given by entry-wise multiplication. The symmetric bilinear map $\hat{e}_6 : G_1^3 \times G_1^3 \rightarrow G_2^6$ is given by

$$\hat{e}_6\left(\begin{pmatrix}a\\b\\c\end{pmatrix},\begin{pmatrix}x\\y\\z\end{pmatrix}\right) \rightarrow \begin{pmatrix}e(a,x) & e(a,y)e(b,x) & e(a,z)e(c,x)\\ 0 & e(b,y) & e(b,z)e(c,y)\\ 0 & 0 & e(c,z)\end{pmatrix}$$

Lemma 1 [13]: Define a map $\mu : Z_p^9 \rightarrow M_2$, $\forall u_1, u_2, u_3 \in M_1$, $\exists \rho_{11}, \rho_{12}, ..., \rho_{33} \in Z_p^9$ and $t_1, ..., t_H \in Z_p$, such that $\prod_{i=1}^{3}\prod_{j=1}^{3} \hat{e}_6(u_i, u_j)^{\rho_{ij}} = 1$, where $\rho_{ij} = \sum_{h=1}^{H} t_h \eta_{hij}$ and $\eta_{hij} \in Z_p$.

For perfect soundness of the NIWI and NIZK proof, the common reference string and simulated reference strings must be computationally indistinguishable, so we also have $\mu(\eta_h) = 1$ for all $\eta_h \in Z_p^9$ and $\eta_{hij} \in \eta_h$ performs as the basis for generating the kernel of μ.

2) Proof Generation: We will use NIWI and NIZK together and/or separately based on different scenarios in our framework. Note that NIWI proof tries to convince that the witnesses in the statement are indistinguishable, where the verifier or adversaries cannot locate which witness (prover) is correspond to the statement, while NIZK proof represents that the statement can be verified without exposing any other information. We denote the credential of A's unique attribute as $x \in G_1$, while there could be other variables in the bilinear paring, i.e., $y \in G_1$. Suppose x and y can form an equation $e(x, y) = T$, where $T \in G_2$. User A uses elements in M_1 and chooses random numbers in Z_p to generate two commitments $Com(x)$ and $Com(y)$. Then, A can make the corresponding NIWI or NIZK proof π based on $Com(x)$ and $Com(y)$. In our scheme, patients/physicians are given the authority to generate unique proofs and commitments for verification.

3) Verification: TA has a bulletin board for publishing the common reference string (crs) used for every member in the system to verify the given proofs. Given proof π_i, $Com(x)$, $Com(y)$, public parameter u_i and the statement $e(x, y) = T$ that we are concerned with, we can set up the following verification equation for a verifier to publicly verify the corresponding statements or equations:

$$\hat{e}_6(Com(x_q), Com(y_q)) = \begin{pmatrix}1 & 1 & 1\\ 0 & 1 & 1\\ 0 & 0 & T\end{pmatrix} \prod_{i=1}^{3} \hat{e}_6(u_i, \pi_i). \quad (1)$$

If the check passes, we can learn that the variables x and y satisfy the statement while ensuring that the verifier learns nothing about x and y.

B. Privacy Level 0-PAAS0

In PAAS0, we are facing the problem of verifying the validity of corresponding credentials issued by RC without exposing them. Without loss of generality, we hereby list the following steps which are implemented in all of our privacy levels.

1) Setup: TA is responsible for generating system parameters and distributing corresponding public/private key pairs to the valid users in the system. We list the possible procedures in the system initiation, although several steps will not be directly implemented in this scenario. In the system setup, TA generates system parameters (include crs) and assigns the ID-based public/private key pairs for each user in the system, and then TA may go offline. For ease of description, we assume that there is a secure channel between TA and any user in the system, like SSL or TSL, which can be achieved by any public key cryptosystems. The key generation procedures are as follows [20]: 1) Input the security parameter ξ to the system and output parameter tuple (p, G_1, G_2, e, g, H). 2) Randomly select a domain master secret $\varsigma \in Z_p^*$ and calculate the domain public key as $P_{pub} = \varsigma g$. Then, TA publishes the domain parameters tuple $(p, G_1, G_2, e, g, H, P_{pub})$ and maintain ς confidential, where $H(\cdot)$ is defined before as $H(\cdot) : \{0, 1\}^* \rightarrow G_1$, and g is a generator of G_1. Given a specific public $ID \in \{0, 1\}^l$, the public/private key (pk_{ID}/sk_{ID}) pair is $H(ID)/\varsigma H(ID)$, which are distributed by TA during the initiation process. We also assign a bunch of collision-resistant pseudonyms for anonymous communication. Taking user A as an example, it will be given a set of pseudonyms, $\mathcal{PS}_A = \{PS_A^{\kappa} | 1 \leqslant \kappa \leqslant |\mathcal{PS}_A|\}$. Each pseudonym of A is given a set of secret keys as $sk_{\mathcal{PS}_A} = \{sk_{\mathcal{PS}_A^\kappa}\} = \{\varsigma_A H(\mathcal{PS}_A^\kappa) \in G_1 | 1 \leqslant \kappa \leqslant |\mathcal{PS}_A|\}$ corresponding to the set of pseudonyms. Note that every party can query TA for public/private key pairs of its pseudonym set and $\varsigma_A \in Z_p^*$ is the master secret selected by TA for A.

2) CRS Generation: The above generation process outputs a set of parameters (p, G_1, G_2, e, g). TA randomly picks $\alpha, \beta, r_u, s_v \leftarrow Z_p^*$. Set $f = g^\alpha, h = g^\beta$, and generate the $u_i \in M_1$, which are $u_1 := (f, 1, g), u_2 := (1, h, g)$ and $u_3 := (f^{r_u}, h^{s_v}, g^{t_w})$, where $t_w = r_u + s_v$. TA sets the crs as $(p, G_1, G_2, e, M_1, M_2, \hat{e}, g, u_1, u_2, u_3, \eta_1, \eta_2, \eta_3)$ and publishes it on the bulletin board for users to verify proofs.

3) Credential Issuance: RC applies the parameters selected by TA, picks a random integer $\tilde{x}_\imath \in Z_p^*$ for a specific class of attribute, where $\imath$ could represents age, affiliation, gender, etc. Once it successfully verifies the specific attributes (via the diagnosis of physicians for patients or certification authority for physicians), it computes the credential $\tilde{v}_\imath = g^{\tilde{x}_\imath} \in G_1$ and $e(g, g) \in G_2$. The public key tuple for the verification is $(g, \tilde{v}_\imath, e(g, g))$, while the secret/sign key held by RC is $\tilde{x}_\imath$ corresponding to different attributes. Intuitively, users can show their validity by presenting $\tilde{v}_\imath$. However, we cannot directly reveal the credential $\tilde{v}$ to the public verifiers, which may incur the impersonation attack when adversaries use credentials to show their validity on specific attributes. For example, to prove user A is a valid physician, we implement the certified signature scheme in [15], [23] for the verification of valid $\tilde{v}$. RC randomly picks group elements $\hat{f}, \hat{h}, \hat{z} \in G_1$, and publishes the tuple $(\hat{f}, \hat{h}, \Gamma)$ as the authority key to verify $\tilde{v}$, where $\Gamma = e(\hat{f}, \hat{z}) \in G_2$ and $\hat{z}$ is a secret key for RC. Then, RC picks a random number $\hat{r} \in Z_p$ and computes $(a, b) := (\hat{f}^{-\hat{r}}, (\hat{h} \cdot \tilde{v})^{\hat{r}} \cdot \hat{z})$. Given $\tilde{v}$, everyone is able to verify the valid credential by checking $e(a, \hat{h} \cdot \tilde{v}) e(\hat{f}, b) = \Gamma$.

4) NIWI *Proof Generation:* As we can see, there are only two variables that we want to hide, $\tilde{v}$ and b. The original schemes [15], [23] is not concerned about the revealing of the credential $\tilde{v}$, while we need to keep those checking processes continuing without exposing the plaintext value of $\tilde{v}$. In this case, A uses the parameters from the published bulletin board $u_1, u_2, u_3 \in M_1$ and chooses $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23} \in Z_p^*$ to commit $\tilde{v}$ and b as follows, $Com(\hat{h} \cdot \tilde{v}) := c_0 := (1, 1, \hat{h} \cdot \tilde{v}) u_1^{r_{11}} u_2^{r_{12}} u_3^{r_{13}}$ and $Com(b) := d_0 := (1, 1, (\hat{h} \cdot \tilde{v})^{\hat{r}} \cdot \hat{z}) u_1^{r_{21}} u_2^{r_{22}} u_3^{r_{23}}$. Apart from this, A also generates a set of NIWI proofs,

$$\pi_i := (1, 1, \hat{f}^{-\hat{r}})^{r_{1i}} (1, 1, \hat{f})^{r_{2i}}. \tag{2}$$

Then, A sends the packet $< c_0, d_0, \pi_1, \pi_2, \pi_3 >$ to the public domain for verification.

5) Public Verification: For privacy concerns, user A may not want to expose his/her real identity or credentials on the third party or the public domain. As result, given the public parameters crs and $(\hat{f}, \hat{h}, \Gamma)$, users can verify the validity of corresponding credentials. Similar to the bilinear map $\hat{e}_6 : G_1^3 \times G_1^3 \to G_2^6$, we define another bilinear map $\hat{e}_9 : G_1^3 \times G_1^3 \to G_2^9$:

$$\hat{e}_9 \left(\begin{pmatrix} a \\ b \\ c \end{pmatrix}, \begin{pmatrix} x \\ y \\ z \end{pmatrix} \right) \to \begin{pmatrix} e(a,x) & e(a,y) & e(a,z) \\ e(b,x) & e(b,y) & e(b,z) \\ e(c,x) & e(c,y) & e(c,z) \end{pmatrix}.$$

Users verify the validity of the corresponding credential by checking the equality of the following equation,

$$\hat{e}_9 \left(c_0, \begin{pmatrix} 1 \\ 1 \\ \hat{f}^{-\hat{r}} \end{pmatrix} \right) \cdot \hat{e}_9 \left(d_0, \begin{pmatrix} 1 \\ 1 \\ \hat{f} \end{pmatrix} \right) \overset{?}{=} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & \Gamma \end{pmatrix} \prod_{i=1}^{3} \hat{e}_9(u_i, \pi_i).$$

Lemma 2 [13]: Let M_1, M_2 be Z_p-modules, for all $r \in Z_p$, $u, u', v \in M_1$, we have $\hat{e}(u^r u', v) = \hat{e}(u, v)^r \hat{e}(u', v)$.

Accordingly, the left hand side of its equation can be derived as follows,

$$\begin{aligned} LHS &= \hat{e}_9 \left(\begin{pmatrix} f^{r_{11}+r_u r_{13}} \\ h^{r_{12}+s_v r_{13}} \\ \hat{h}\tilde{v} \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \hat{f}^{-\hat{r}} \end{pmatrix} \right) \\ &\cdot \quad \hat{e}_9 \left(\begin{pmatrix} f^{r_{11}+r_u r_{13}} \\ h^{r_{12}+s_v r_{13}} \\ g^{\dot{r}} \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \hat{f}^{-\hat{r}} \end{pmatrix} \right) \\ &\cdot \quad \hat{e}_9 \left(\begin{pmatrix} f^{r_{21}+r_u} \\ h^{r_{22}+s_v} \\ (\hat{h}\tilde{v})^{\hat{r}} \hat{z} \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \hat{f} \end{pmatrix} \right) \cdot \hat{e}_9 \left(\begin{pmatrix} f^{r_{21}+r_u} \\ h^{r_{22}+s_v} \\ g^{\ddot{r}} \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \hat{f} \end{pmatrix} \right) \end{aligned}$$

Due to the page limit, we only show the verification of the intersection of row 3 and column 3 of the corresponding matrix,

$$\begin{aligned} LHS_{33} &= e(\hat{h}\tilde{v}, \hat{f}^{-\hat{r}}) e(g^{\dot{r}}, \hat{f}^{-\hat{r}}) e((\hat{h}\tilde{v})^{\hat{r}}, \hat{f}) e(\hat{z}, f) e(g^{\dot{r}}, f) \\ &= e(\hat{z}, f) e(g, f)^{\ddot{r} - \hat{r}\dot{r}} \end{aligned}$$

where $\dot{r} = r_{11} + r_{12} + t_w r_{13}$ and $\ddot{r} = r_{21} + r_{22} + t_w r_{23}$. Meanwhile, the right hand side of the equation can be derived as follows,

$$\begin{aligned} RHS_{33} &= \Gamma e(g, f^{-\hat{r} r_{11} + r_{21}}) e(g, f^{-\hat{r} r_{12} + r_{22}}) e(g, f^{-\hat{r} r_{13} + r_{23}}) \\ &= \Gamma e(g, f)^{-\hat{r}\dot{r} + \ddot{r}}. \end{aligned}$$

It is obvious that if the provided credential is verified, the above two equations will be equal, which implies that the attribute is verified by RC while keeping it undisclosed. Note that patients attributes also can be verified using PAAS0.

C. Privacy Level 1-PAAS1

In most cases, patients have different values on a specific attribute, like the length of disease history, the size of cancer or the amount of remedy, which are highly concerned with privacy issues and easily to be identified. In PAAS1, we consider the scenario where there is a server or physicians who need to verify the value of the corresponding attributes of patients but without requiring patients' real identities. Based on different values on an attribute, RC issues distinct certificates to different users. For the privacy concerns, patients would not turn in the certificate that RC issued on specific value of an attribute for verification. Note that certificates for the same attribute value on different real identities are the same, which implies when patients are trying to verify the similarity of their ages, instead of comparing the real value of an attribute, what we need is only to verify the NIWI or NIZK proof based on the commitments of their corresponding certificates.

1) Certificate Issuance: We consider an asymmetric setting between the server S (we use server to represent possible physicians or organizations that need to verify patients attribute values) and an individual patient. We will use Boneh-Boyen signature scheme [22] to sign the attribute value, which is secure under weak chosen message attacks based on the q-Strong Diffie-Hellman (q-SDH) assumption.

Issuance: RC continues to use $\tilde{x}_\imath$ to sign each specific value of corresponding attribute. We assume the value of attribute in our system could be represented as a message $m_{\imath,\jmath} \in Z_p$, where $\imath$ denotes the general classification of attributes and $\jmath$ is the specific value of those attribute, like *Stage II* represents the "degree of severity of AIDS" (a kind of an attribute). To prevent the malicious attack from patients (discussed in later sections), we add a random number $\varepsilon \in Z_p \backslash \{-(\tilde{x}_\imath + m_{\imath.\jmath})\}$ selected by RC for the purpose of update for the same attributes and security of verification. Given the message $m_{\imath.\jmath}$ along with the secret key $\tilde{x}_\imath$ and ε, RC outputs a certificate $\sigma_{\imath.\jmath} := g^{1/(\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon)} \in G_1$. Besides, RC issues the server S an additional certificate $\delta^{\varepsilon}_{i.j} := g^{(m_{\imath.\jmath}+\varepsilon)}$ for further verification. Note that RC periodically updates $\sigma_{\imath.\jmath}$ for patients accompanied with $\delta^{\varepsilon}_{i.j}$ for the server S.

Verification: Given the additional certificate $\delta^{\varepsilon}_{i.j}$ from the server S, patients need to use $\tilde{v}_\imath$ and $\sigma_{\imath.\jmath}$ to prove the equation $e(\sigma_{\imath.\jmath}, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon)}) e(g^{-1}, g) = 1$ which implies that the value of the attribute that he/she holds is the same as that on the server side. Based on the NIWI proof system, the server would not be able to get the plaintext of $\sigma_{\imath.\jmath}$, which perfectly hides the certificate privacy of patients. If the equation is not satisfied, the patient learns nothing about the correct additional credential from S while the server S cannot obtain the certificate $\sigma_{\imath.\jmath}$.

2) NIWI *Proof Generation and Verification:* We also utilize the proof generation process in PAAS0 used to verify the validity of the attribute. Apart from that, we also need to generate the NIWI proofs for proving the equality of values of a specific attribute. Before the NIWI proof process begins, a patient (which could be seen as a prover) needs to obtain the additional certificate from the server side to generate commitments and proofs. Suppose S accepts the query from patient A,

1. $\mathcal{PS}^{\kappa}_A \to S : E_{pk_S}(m_{\imath.\jmath}), \tau_1, SIG(E_{pk_S}(m_{\imath.\jmath}||\tau_1)$
2. $S \to \mathcal{PS}^{\kappa}_A : E_{pk_{\mathcal{PS}^{\kappa}_A}}(\delta^{\varepsilon}_{i.j}), \tau_2, SIG(E_{pk_{\mathcal{PS}^{\kappa}_A}}(\delta^{\varepsilon}_{i.j}||\tau_2)$

where $E(\cdot)$ is the ID-based encryption and SIG denotes the efficient ID-based signature scheme [24] which could be verified by the corresponding public ID. Note that τ represents the timestamp used to prevent reply attacks.

The following NIWI proof process includes three steps: committing to the variables, generating proof and verification.

Committing to the variables: We need to give NIWI proofs for the following equations. We also list the validity verification equation in PAAS0 to denote that both of the equations must be simultaneously satisfied and verified by the verifier:

$$e(a, \hat{h} \cdot \tilde{v}_\imath) e(\hat{f}, b) = \Gamma \tag{3}$$

$$e(\sigma_{\imath.\jmath}, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon)}) e(g^{-1}, g) = 1. \tag{4}$$

Patient A chooses the $u_i \in M_1$ from the published crs to commit those variables as follows, taking the second equation as an example. A first chooses the elements $u_1, u_2, u_3 \in M_1$ from crs and randomly selects $r_1, r_2, r_3, r'_1, r'_2, r'_3 \leftarrow Z_p$. Then, A computes $Com(\sigma_{\imath.\jmath}) := c_1 = (1, 1, \sigma_{\imath.\jmath}) u_i^{r_i} = (f^{r_1 + r_3 r_u}, h^{r_2 + r_3 s_v}, \sigma_{\imath.\jmath} g^{r_1 + r_2 + r_3(r_u + s_v)})$, and A also commits to the other variable as $Com(\tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon)}) := d_1 = (1, 1, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon)}) u_i^{r'_i} = (f^{r'_1 + r'_3 r_u}, h^{r'_2 + r'_3 s_v}, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon) + r'_1 + r'_2 + r'_3(r_u + s_v)})$.

Generating NIWI **proof:** For patient A, it needs to generate the NIWI proof $\bar{\pi}_i$ by using the parameters in crs and the commitments. Given the kernel vectors of μ_6 in crs, $\eta_1 := (0, 1, 0, -1, 0, 0, 0, 0, 0)$, $\eta_2 := (0, 0, 1, 0, 0, 0, -1, 0, 0)$, $\eta_3 := (0, 0, 0, 0, 0, 1, 0, -1, 0)$, A randomly selects $t_1, t_2, t_3 \in Z_p$ to generate the NIWI proofs as follows:

$$\bar{\pi}_i = \prod_{j=1}^{3} u_j^{\sum_{h=1}^{3} t_h \eta_{hij}} (1, 1, \sigma_{\imath.\jmath})^{r'_i} d_1^{r_i}$$

Verification: When patient A wants to verify himself that he has the corresponding credential which satisfy the requirements of the server, A sends the packet $< c_1, d_1, \bar{\pi}_1, \bar{\pi}_2, \bar{\pi}_3 >$ to the server. We can modify the equation (4) to $e(\sigma_{\imath.\jmath}, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon)}) = e(g, g)$, where $T = e(g, g)$ is a known factor. After obtaining proofs along with commitments, the server can verify the validity of the user's attribute according to equation (1):

$$LHS = \hat{e}_6 \left(\begin{pmatrix} f^{r_1 + r_3 r_u} \\ h^{r_2 + r_3 s_v} \\ \sigma_{\imath.\jmath} g^{\hbar} \end{pmatrix}, \begin{pmatrix} f^{r'_1 + r'_3 r_u} \\ h^{r'_2 + r'_3 s_v} \\ \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon) + \hbar'} \end{pmatrix} \right) \tag{5}$$

where $\hbar = r_1 + r_2 + r_3(r_u + s_v)$ and $\hbar'$ denotes the $r'_1 + r'_2 + r'_3(r_u + s_v)$, respectively. The server S then checks the right hand side as follows,

$$\begin{aligned} \prod_{i=1}^{3} \hat{e}_6(u_i, \bar{\pi}_i) &= \prod_{i=1}^{3} \hat{e}_6(u_i, (1, 1, \sigma_{\imath.\jmath})^{r'_i} d_1^{r_i}) \\ &= \hat{e}_6 \left(\prod_{i=1}^{3} u_i^{r'_i}, (1, 1, \sigma_{\imath.\jmath}) \right) \hat{e}_6 \left(\prod_{i=1}^{3} u_i^{r_i}, d_1 \right) \\ &= \hat{e}_6 \left(\begin{pmatrix} f^{r'_1 + r'_3 r_u} \\ h^{r'_2 + r'_3 s_v} \\ g^{\hbar'} \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \sigma_{\imath.\jmath} \end{pmatrix} \right) \\ &\cdot \hat{e}_6 \left(\begin{pmatrix} f^{r_1 + r_3 r_u} \\ h^{r_2 + r_3 s_v} \\ g^{\hbar} \end{pmatrix}, \begin{pmatrix} f^{r'_1 + r'_3 r_u} \\ h^{r'_2 + r'_3 s_v} \\ \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon) + \hbar'} \end{pmatrix} \right) \end{aligned}$$

Note that we omit several terms in the proof $\bar{\pi}_i$ since we have $\prod_{i=1}^{3} \prod_{j=1}^{3} \hat{e}_6(u_i, u_j)^{\rho_{ij}} = 1$ according to *Lemma 1*. By directly applying the bilinear operation to the above equations, we can easily check that all the entries satisfy the equalities in the equation (5) except the last one in the matrix. However, we observe the result in the row 3 and column 3 of the corresponding matrix. We expand all the pairing results as follows,

$$\begin{aligned} LHS_{33} &= e(\sigma_{\imath.\jmath} g^{\hbar}, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon) + \hbar'}) \\ &= e(g, g)^{\left(\frac{1}{\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon} + \hbar \right)(\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon + \hbar')} \\ &= e(g, g)^{1 + \hbar(\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon) + \frac{\hbar'}{\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon} + \hbar\hbar'}, \end{aligned} \tag{6}$$

and according to equation (1),

$$\begin{aligned} RHS_{33} &= T \cdot e(g^{\hbar'}, \sigma_{\imath.\jmath}) e(g^{\hbar}, \tilde{v}_\imath \cdot g^{(m_{\imath.\jmath}+\varepsilon) + \hbar'}) \\ &= e(g, g)^{1 + \frac{\hbar'}{\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon} + \hbar(\tilde{x}_\imath + m_{\imath.\jmath} + \varepsilon + \hbar')} \end{aligned} \tag{7}$$

the two results are identical.

After the verification process, the server S is able to convince itself that patient A holds the attribute that is satisfied S's requirement: the value of the attribute provided by A is the same as S's. Thus, S can give the access privilege or other rights based on the verification results. Note that PAAS1 also works well under the symmetric scenario where the privacy of the attribute value is not a must, where patients or physicians can mutually verify their corresponding specifications. Our implementation in PAAS1 specifies a special case of asymmetric structure where the value of the attribute is crucial for the application.

*D. Privacy level 2-*PAAS2

We have formally described our scheme on the PAAS0 and PAAS1, where patients allow to leak several information to the server or physicians. For computational efficiency, we leverage the server side to share an additional attribute certificate for the verification process. However, according to the privacy definition in PAAS2, patients cannot disclose any private information to a stranger or even a so-called trusted server if we are not sure whether they share the same attribute values as theirs. The major difference between previous two privacy levels and PAAS2 is that PAAS2 hides attributes from the verifier when the verifier does not hold those attributes and the corresponding values. The most possible scenario for PAAS2 exists in patient-patient interactions, where patients with same attributes and values may want to communicate. However, what patients most concerned about is they do not want to reveal the identity and disease detail to patients who do not have the same attribute values. We can leverage pseudonyms as a solution to the exposure of identity privacy, and design the scheme to address other privacy concerns.

1) Initiation: Contrary to PAAS1, we do not make additional certificate $\delta^{\varepsilon}_{\imath.\jmath}$ on the server side. Instead, the communication parties in PAAS2 are in a symmetric fashion, in the sense that both patients are able to obtain the same attribute set when the protocol ends. RC issues the valid user certificates corresponding to the specific attribute as $\bar{\sigma}_{\imath.\jmath} := g^{1/(\tilde{x}_\imath + m_{\imath.\jmath})}$. Two patients will mutually authenticate each other using the NIWI proofs to show they have a valid credential and verifiable value on the attribute,

1. $\mathcal{PS}^{\kappa}_{A} \rightarrow \mathcal{PS}^{\kappa'}_{B} : c_{\imath.\jmath}, d_{\imath.\jmath}, \pi_1, \pi_2, \pi_3,$
2. $\mathcal{PS}^{\kappa'}_{B} \rightarrow \mathcal{PS}^{\kappa}_{A} : c'_{\imath.\jmath}, d'_{\imath.\jmath}, \pi'_1, \pi'_2, \pi'_3.$

where $c_{\imath.\jmath}$ denotes $Com(\bar{\sigma}_{\imath.\jmath})$, and $d_{\imath.\jmath}$ represents $Com(\tilde{v}_\imath \cdot g^{m_{\imath.\jmath}})$, respectively. Two users can mutually verify each other using the NIWI proofs based on equation (3) and (4). Note that both sides can obtain the information of the compared attributes from the above process.

2) Equality NIWI *proof generation:* The above authentication process can verify the validity of the corresponding attributes and values, but it does not imply any relationship between the attribute and identities. For the case of checking the intersection set of the compared attributes, we require the two patients to send the random parameters to each other, which brings out a new series of NIWI proofs $\dot{\pi}_i$ to verify the equality of two committed values.

Suppose A uses $s_i, s'_i \in Z_p$ to commit $\bar{\sigma}_{\imath.\jmath}$ and $\tilde{v}_\imath \cdot g^{m_{\imath.\jmath}}$, respectively. Also, B chooses $t_i, t'_i \in Z_p$ to generate $c'_{\imath.\jmath}$ and $d'_{\imath.\jmath}$. For generating equality NIWI proofs $\dot{\pi}_i$, B sends back to A the random parameter set $\{t'_i\}$. Then, A computes the following results and sends to B,

$$\dot{\pi}_i = \prod_{j=1}^{3} u_j^{\sum_{h=1}^{3} t_h \eta_{hij}} (1, 1, \bar{\sigma}_{\imath.\jmath})^{t'_i} (d'_{\imath.\jmath})^{s_i}$$

Note that sharing the random parameter set $\{t'_i\}$ will not reveal the corresponding credentials. Although A is able to generate a set $(1, 1, 1)u_i^{t'_i}$, it is infeasible to recover the credential from $d'_{\imath.\jmath}$ and the above value set.

3) Verification: The verification process is more or less similar to that in PAAS1. When A sends the equality NIWI proofs $\dot{\pi}$ to B, B checks the equality of the following equation:

$$\hat{e}_6(c_{\imath.\jmath}, d'_{\imath.\jmath}) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & T \end{pmatrix} \prod_{i=1}^{3} \hat{e}_6(u_i, \dot{\pi}_i). \tag{8}$$

which implies that $e(g, g)^{\frac{x_\imath + m^B_{\imath.\jmath}}{x_\imath + m^A_{\imath.\jmath}}} = e(g, g)$, where $m^A_{\imath.\jmath}$ and $m^B_{\imath.\jmath}$ are the attribute values of A and B on the same attribute. If the above equation is satisfied, B is convinced the equality of two attribute values. Otherwise, B will find the inequality of the equation (8), which shows that the attributes of A and B are not identical. Apart from this, both A and B learn nothing about credentials and attribute values from each other. B can further verify itself to A reversely. Here, we only show the verification process for one single attribute value. It is obvious that we can apply this scheme in verifying a set of attributes by repeating the same processes. When the protocol ends, both patients will only learn the identical attribute values that they have. However, for a single user, it will incur more pairing invocations than PAAS1 for one attribute.

*E. Privacy Level 3-*PAAS3

PAAS3 requires that two users can mutually obtain the cardinality of the intersection of attribute set (defined as similarity score) without learning the detail of each identical attribute value. Compared to PAAS2, users even would not want to disclose the attribute value to the one that has the same value as theirs. They only can accept the comparison results on the number of identical attribute values in the intersection set. Here, we assume the weight of each attribute value is the same for general perspective. PAAS3 not only relies on the NIWI proofs introduced in the previous subsection, but also applies the NIZK proofs to prove the equality of the corresponding ciphertexts and commitments without exposing the real plaintext value. Then, each user implements homomorphic encryption on the received encrypted packet. Finally, both users are able to decrypt the values, where 1 denotes the same attributes, and otherwise represents different attribute values between two users. By randomly rearranging the sequence of the homomorphic results, no one is able to obtain the detail of which two attributes are identical when we apply a large attribute set.

The NIWI proof generating and verification processes for the two involved users are the same in the first step of PAAS2. What A wants to compare is a set of attributes $\mathcal{S}^*_A$, and we further assume all $m^A_{\imath.\jmath} \in \mathcal{S}^*_A$ have been verified using the technique listed in the previous subsection. We make the following modification and addition for PAAS3.

1) Selective-tag encryption: In general, PAAS3 needs to perform operations on the encrypted value instead of commitments, because we cannot perform the operations over the committed values, such as comparison, addition, multiplication, etc. The reason that we choose selective-tag encryption [25] in our scheme is that it has an encrypted form which perfectly matches the commitment scheme used in the NIWI proof. We briefly review the selective-tag encryption scheme,

Parameter Generation: TA assigns a tuple $\bar{pk} := (\bar{f}, \bar{h}, \bar{k}, \bar{l}) \in G_1^4$ as a public key for a user, and distributes the private key (χ, ψ) to a user where $\bar{f} = g^\chi, \bar{h} = g^\psi$.

Encryption: User chooses random numbers $\bar{r}, \bar{s} \in Z_p$ and select a public tag $\bar{t}$ to encrypt a message $\bar{m}$ as $\mathcal{E}(\bar{m}) := (\bar{f}^{\bar{r}}, \bar{h}^{\bar{s}}, g^{\bar{r}+\bar{s}}\bar{m}, (g^{\bar{t}}\bar{k})^{\bar{r}}, (g^{\bar{t}}\bar{l})^{\bar{s}})$.

Decryption: Decryption can be done by computing $\bar{m} = g^{\bar{r}+\bar{s}}\bar{m} \cdot (\bar{f}^{\bar{r}})^{-1/\chi}(\bar{h}^{\bar{s}})^{-1/\psi}$.

2) NIZK *proof generation:* The generation process of NIZK is similar to NIWI in [15], where we commit to the exponential of the generator g. We first show the statements that we need to prove. Given the ciphertext of encrypted certificate of A as $\mathcal{E}(\sigma^A_{\imath.\jmath}) := (X_1, X_2, X_3, X_4, X_5) = (f^{\bar{r}_e}, h^{\bar{s}_e}, g^{\bar{r}_e+\bar{s}_e}\sigma^A_{\imath.\jmath}, (g^{\bar{t}}\bar{k})^{\bar{r}_e}, (g^{\bar{t}}\bar{l})^{\bar{s}_e})$. According to the previous subsections, $Com(\sigma^A_{\imath.\jmath}) :=$

$(C_1, C_2, C_3) = (f^{r_1+r_3r_u}, h^{r_2+r_3s_v}, \sigma^{A}_{\imath.\jmath} g^{r_1+r_2+r_3(r_u+s_v)})$. Setting $r_0 = r_1 - \bar{r}_e$ and $s_0 = r_2 - \bar{s}_e$, we have the following statements that need to be simultaneously satisfied for proving the equality of the committed value and the encrypted value,

$$\varphi = 1 \wedge (C_1^{-1}X_1)^{\varphi} f^{r_0} (f^{r_u})^{r_3} = 1 \wedge (C_2^{-1}X_2)^{\varphi} h^{s_0} (h^{s_v})^{r_3} = 1$$
$$\wedge (C_3^{-1}X_3)^{\varphi} g^{r_0+s_0} (f^{r_u+s_v})^{r_3} = 1$$

where we cannot disclose φ, r_0, s_0, r_3 to the verifier and we need to prove the above equations given the ciphertexts and the commitments. Note that $\wedge$ denotes *and*.

To generate the commitments, we randomly select two numbers $\theta, \zeta \in Z_p$ and add one more universal parameter $u := u_1^{\theta} u_2^{\zeta}$ onto crs, which is linearly independent of u_1 and u_2. Taking second equation as an example, we need to commit to the exponentials φ, r_0, t as $\mathcal{C}_1 = u^{\varphi} u_1^{\nu_1} u_2^{\nu_2}$, $\mathcal{C}_2 = u^{r_0} u_1^{\nu_1} u_2^{\nu_2}$ and $\mathcal{C}_3 = u^{t} u_1^{\nu_1} u_2^{\nu_2}$,where $\nu_i \in Z_p$ are random numbers. Then, A can generate NIZK proof as follows,

$$\ddot{\pi}_i = (1, 1, C_1^{-1}X_1)^{\nu_i} (1, 1, f)^{\nu_i} (1, 1, f^{r_u})^{\nu_i}$$

Since A has already sent the commitment of $\sigma^{A}_{\imath.\jmath}$ to B, user B is able to compute $C_i^{-1}X_i$ after A delivers the ciphertexts.

3) Verification: Due to page limit, we do not give the detail of the checking process. However, the process is similar to eq(3),

$$\prod_{i=1}^{3} \hat{e}_9 \left(\begin{pmatrix} 1 \\ 1 \\ Y_i \end{pmatrix}, \mathcal{C}_i \right) = \hat{e}_9 \left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, u \right) \prod_{i=1}^{2} \hat{e}_9(\ddot{\pi}_i, u_i) \qquad (9)$$

where Y_i is each of the elements in the set $(C_1^{-1}X_1, f, f^{r_u})$, respectively. The user B checks the satisfiability of all the four equations. If the check passes, B is convinced that the value in the commitment is equal to that in the ciphertext.

4) Deriving Similarity Score: The similarity score between two users is the size of identical attributes of their intersection sets. Based on the ciphertexts of certificates, users perform homomorphic encryption on each single ciphertext and return to each other for deriving the result.

Homomorphic Encryption: Our proposed scheme relies on the encryption results of selective-tag encryption which has the following properties,

1. Multiplicative homomorphic property:

$$\begin{aligned} \mathcal{E}(\overline{m}_1)\mathcal{E}(\overline{m}_2) &= (f^{\overline{r}_1+\overline{r}_2}, h^{\overline{s}_1+\overline{s}_2}, g^{\overline{r}_1+\overline{r}_2+\overline{s}_1+\overline{s}_2}\overline{m}_1 \cdot \overline{m}_2, \\ &\quad (g^{\overline{t}}\overline{k})^{\overline{r}_1+\overline{r}_2}, (g^{\overline{t}}\overline{l})^{\overline{s}_1+\overline{s}_2}) \\ &= \mathcal{E}(\overline{m}_1 \cdot \overline{m}_2) \end{aligned}$$

2. Self blinding property:

$$\mathcal{D}(\mathcal{E}(\overline{m}, \overline{r}_1, \overline{s}_1, \overline{t})) = \mathcal{D}(\mathcal{E}(\overline{m}, \overline{r}_1 + \overline{r}_2, \overline{s}_1 + \overline{s}_2, \overline{t})).$$

Score Derivation: We suppose both A and B mutually give a same set of attribute classes for the comparison, like $\mathbb{O}$:={*affiliation, height, ages, disease,...*} with a predefined order. For the verifier, it is impossible to guess out the values inside the commitment other than verifying the satisfiability of the corresponding statements. Thus, when A gives to B a set of commitments and the ciphertext, it should tell the order of the attribute order that it wants to prove. Then, B can bind encrypted values to the received ciphertexts,

1. $PS_A^{\kappa} \rightarrow PS_B^{\kappa'} : \{\mathcal{E}_{\bar{pk}_A, \bar{t}_A}(\sigma^{A}_{\imath.\jmath}) | \imath \in \mathcal{S}_A^*\}^{\mathbb{O}}$
2. $PS_B^{\kappa'} \rightarrow PS_A^{\kappa} : \{\mathcal{E}_{\bar{pk}_A, \bar{t}_A}(\sigma^{A}_{\imath.\jmath}) \cdot \mathcal{E}_{\bar{pk}_A, \bar{t}_A}((\sigma^{B}_{\imath.\jmath})^{-1})\}^{\mathbb{R}}$

where $\mathbb{R}$ denotes the random permutation of the corresponding set. After receiving the encrypted credential set, B performs the selective-tag encryption on $(\sigma^{B}_{\imath.\jmath})^{-1} := g^{-1/(x_\imath + m^{B}_{\imath.\jmath})}$ by using the public key of PS_A^{κ} and the tag according to the predefined order $\mathbb{O}$. Then, B multiplies its own encryption together with the received encrypted set with order of $\mathbb{O}$. Randomly permuting the generated encrypted set provides the randomness in determining two identical attributes among the whole intersection set.

Finally, A uses the private key (χ, ψ) to decrypt the ciphertext in the set one by one. What A needs to do is to collect the number of "1" in the set, which implies the similarity score between A and B on specified sets $\mathcal{S}_A^*$ and $\mathcal{S}_B^*$. Otherwise, decryption results are arbitrary bit strings due to the inequality of $\sigma^{A}_{\imath.\jmath}$ and $\sigma^{B}_{\imath.\jmath}$.

V. Performance Analysis

This section studies on how the security requirements are achieved in PAAS for eHealth networks based on the objective and adversary model defined before, and how to enhance the resilience of PAAS to the general attacks and specific attacks to different privacy levels. The efficiency of the proposed system in terms of computation load is also discussed.

A. Security Analysis

1) Identity and Attribute Privacy: Since we render rooms for patients and physicians to use frequently changed pseudonyms to communicate with each other, the real identity is hidden from being traced. For the attribute privacy, the verification and comparison processes only use commitments of corresponding attributes and values. Without directly exposing plaintexts of such attributes, we ensure that adversaries cannot obtain detailed information about them. Although in PAAS2, patients obtain the identical attributes values from each other based on user's intentions, it still preserves the privacy for the distinct attribute values from being leaked. One of the most important privacy requirements is the unlinkability between identities and attribute value set. Since our scheme applies randomness on both commitments of attribute values and pseudonyms, adversaries are not able to find the linkage between identities and attributes. Taking a step further, commitments generated from the same attributes are distinct, which fundamentally prevents adversaries from obtaining attribute values of particular users.

2) Countermeasures to possible attacks: We first give the security analysis on general attacks that all of the privacy levels will face. Then, we will discuss possible privacy leakages and attacks launched in each of the four levels.

Tracing Attacks: Tracing attacks take place when adversaries collect enough privacy information to link a particular real identity. Our scheme perfectly prevents this kind of attack by using random numbers in commitments and proofs. First of all, the adversary needs to be a valid user in the system, otherwise, its proofs cannot be verified by the other side. Second, even if the authentication process passes, a user will not keep using the same random numbers to generate commitments. According to the previous analysis, the adversary cannot trace any user because they are unable to establish the linkage between identities and attribute set.

Collusion Attacks: Collusion attack is a very powerful attack and will severely threaten the security and privacy requirements of our scheme if not thwarted. According to our assumptions, collusion attacks can happen among users or between users and RC. The first type of collusion attack can be easily prevented. If a group of malicious users want to find out the privacy issues (like attributes) of a particular user, they have to launch numbers of queries to this user. However, the user generates different commitments which may represent the same attributes by using different pseudonyms, attackers cannot tell the relationship between commitments and pseudo-identities. Taking a step further, the assumption of the NIWI system requires the composable witness indistinguishability, the adversary even cannot distinguish a real crs from a simulated crs. On a simulated crs, it is perfectly indistinguishable which witness the prover used, in the sense that a verifier has no knowledge of the credential the prover uses. On the other hand, as a semi-trusted RC, even if a malicious user colludes with it, our scheme provides solution to thwart. We assume the purpose of malicious users is to find out a target user's attribute value. If the authentication process passes, the adversary will hold the commitment of $\sigma_{\imath.\jmath}$ of the target user. Although the adversary is able to obtain all attributes of any user in the system from RC, it cannot find the linkage between the plaintext in RC and $Com(\sigma_{\imath.\jmath}) := (f^{r_1+r_3r_u}, h^{r_2+r_3s_v}, \sigma_{\imath.\jmath}g^{r_1+r_2+r_3(r_u+s_v)})$ due to the fact that only the target user knows random numbers r_i composing the corresponding commitments. Even adversaries launch the statistical attack together with RC, they cannot obtain credentials nor certificate since it is impossible for RC to tell the relationships among multiple commitments.

Attacks on PAAS0: Since PAAS0 only consider the validity of attributes, the most possible attack is impersonation attack by using others' credentials. However, because end users use pseudonyms, the only possible way is to call for TA to perform as an arbiter who can open the commitment and to trace back users who leak their credentials. Suppose we have a suspected user with a commitment $Com(b)$, TA can use the system parameter (α, β) to open the commitment as follows: $(f^{r_1+r_3r_u})^{-1/\alpha} \cdot (h^{r_2+r_3s_v})^{-1/\beta} \cdot b \cdot g^{r_1+r_2+r_3(r_u+s_v)} = b$. Together with the pseudonym that this user used, we can locate the real identity of the adversary. Another possible attack that may compromise the attribute privacy of a user is what we call the "unique identification" attack. For example, if an attribute of physician is the only one who has been verified by a particular professional authority ($\tilde{x}_\imath$ is unique), directly revealing $\Gamma_\imath$ for public verification discloses the identity in case of someone knows there is only one physician (with that particular attribute) who works at that hospital. To avoid this kind of attack, our system requires all users (physicians and patients) should have a same cardinality of their attribute set, which denotes the value of the public parameter Γ remains stable for all physicians and patients. Thus, adversaries cannot identify a particular user according to different Γ.

Attacks on PAAS1: Due to the asymmetric structure of PAAS1, there are two types of attacks that could happen, which are "self-proving" and certificate leakage. For self-proving, a user may use the credential issued by RC and its own attribute $m_{\imath.\jmath}$ to prove the equation (3) and (4). To eliminate this kind of attacks, we ask RC to give an additional certificate to the server side, which is $\delta^{\varepsilon}_{\imath.\jmath} := g^{(m_{\imath.\jmath}+\varepsilon)}$. Since the adversary is not aware of ε, it cannot generate the equation (4) by itself except using the additional certificate from the server. As another aspect of "self-proving" attack, malicious users may collude together to obtain the random number ε, which enables them to generate their own additional certificate in order to avoid the verification process. However, due to our assumption that the DLP is hard, adversaries cannot figure out the value of $m_{\imath.\jmath} + \varepsilon$ based on $\delta^{\varepsilon}_{\imath.\jmath}$. Also, once the misbehavior is detected, we can use the countermeasures in PAAS0 to trace back and revoke the access privilege of the malicious users when they request the additional certificate from the server S. We further render RC and designated server S the ability to periodically update the random number ε, which hinders the revoked user to further utilize the previous additional certificate for the authentication. We note that we cannot prevent collusion attacks with RC in PAAS1, because RC will directly tell attribute values together with the additional certificate once adversaries have already obtained it. However, in several kinds of scenarios, such as checking Boolean type attributes, we may not care about leaking the server of privacy (i.e., a doctor's affiliation). It is still acceptable to implement such scheme with the purpose of providing privacy for users, not the server. Certificate leakage is impossible also according to the assumption that the DLP problem is hard.

Attacks on PAAS2: In PAAS2, each user mutually sends back random numbers used in generating commitments for equality check. We consider two major types of attacks that may potentially expose attribute values from both sides. The first type of attack comes from sending random numbers used for verification. For example, user B sends back to A the random number set $\{t'_i\}$ for A to generate $\dot{\pi}_i$. It is obvious that A can use the received random number set to construct $(1, 1, \tilde{v}_\imath)u_1^{t'_1}u_2^{t'_2}u_3^{t'_3} := (f^{t'_1+t'_3r_u}, h^{t'_2+t'_3s_v}, \tilde{v}_\imath^A \cdot g^{t'_1+t'_2+t'_3(r_u+s_v)})$. Comparing to the received $d'_{\imath.\jmath} := (f^{t'_1+t'_3r_u}, h^{t'_2+t'_3s_v}, \tilde{v}_\imath^B \cdot g^{m^B_{\imath.\jmath}+t'_1+t'_2+t'_3(r_u+s_v)})$, it is difficult to derive $g^{m^B_{\imath.\jmath}}$ from the above values due to the assumption that DLIN problem is hard, which guarantees the attribute privacy when two users sends random numbers back and forth. The other possible attack performs on the outcome of comparison results. According to the requirements of PAAS2, users learns nothing if their attribute values are not identical. The verification results returns 1 if and only if $e(g, g)^{(x_\imath+m^B_{\imath.\jmath})/(x_\imath+m^A_{\imath.\jmath})} = e(g, g)$. Otherwise, the result outputs an arbitrary string instead of explicit listing the values of $m^A_{\imath.\jmath}$ and $m^B_{\imath.\jmath}$. It is also possible for users returning incorrect random numbers and/or proofs, which renders the inconsistent comparison results on two sides. However, our system provides the arbitration algorithm which allows TA to be the arbiter and prevents from taking the risk.

Attacks on PAAS3: The privacy level 3 requires patients to mutually obtain a number of identical attributes instead of the detail of the intersection set. We have explained that the statistical attacks are infeasible for adversaries since a responser uses different pseudonyms to communicate. We further prohibit a user to respond different queries in one time slot, or the colluded users may obtain the target user's attribute value set based on the statistical results. Another type of attack is on the inconsistency of the similarity score, where users do not honestly encrypt and multiply the valid $\sigma^{-1}_{\imath.\jmath}$ on the received packet. There are two possible countermeasures for the inconsistency of the comparison results. Firstly, we can apply another set of NIWI proofs that checking the validity and verifiability of $\sigma^{-1}_{\imath.\jmath}$. Then, we can utilize the homomorphic encryption to guarantee the consistency of the similarity score. Second, another possible countermeasure comes from [6] which encrypts random numbers used in generating the ciphertext and sends to the other end user for checking the consistency of the result. Due to page limit, we refer [6] for the detail which is applicable to our scheme.

B. *Efficiency Analysis*

Due to the page limit, we only consider the computational cost in our proposed framework, and we will further discuss the communication and storage cost in our full version paper. We used the Pairing-Based Cryptography (PBC-0.5.8) Library to implement our simulation. We take Tate pairing as our basic pairing operation. The elliptic curve we use for the our scheme is type A. A curve of such type has the form of $y = x^3 + x$. The order of the curve is around 160 bits, as is F_p, the base field. For the experiments, we use MacBook Pro with an Intel Core 2 Duo 2.8GHz and 4GB RAM. For the simplest case in PAAS0, to commit two variables consists of 6 group elements and the verification process costs 3 group operations. In PAAS1, a user needs 9 group elements to commit 3 variables, while the server side will incur 18 pairing operations for the verification. We also have ID-based encryption and decryption process in PAAS1, which incurs at most one pairing computation in each of the encryption, decryption, signature generating and verification processes, respectively, according to [20], [24]. So, the proofs in PAAS1 consists of 35 group elements, in which a user needs to calculate 13 pairing operations. For the authentication between two patients in PAAS2, it will have 54 group operations. In addition, the user needs 9 group elements for checking the equality of credentials on each side. Thus, PAAS2 will incur approximately 72 group operations in total. We simply consider one attribute situation in PAAS3 for simplicity. With the additional 15 group elements in NIZK and 5 on the ciphertext, PAAS3 totally consists of 94 group elements. Comparing to the the time spent on the computation of group elements, we omit the other kind of computational cost in PAAS3. All the timing reported are averaged over 100 randomized runs. Our results on the cases of more than one attribute will be provided in the full version of this work.

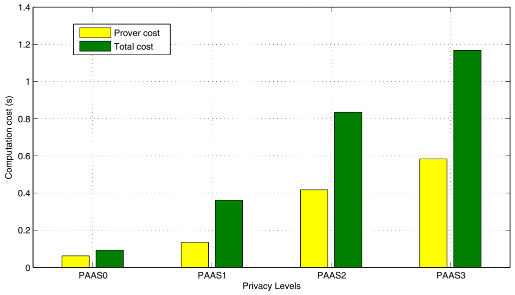

Fig. 2. Computation cost for single attribute.

VI. Conclusion

In this paper, we propose a framework of privacy-preserving attribute-based authentication system in eHealth networks. Our framework applies the non-interactive proof system as the basic building block, in which we give formal definitions of four progressive privacy levels. The attribute-based authentication schemes designed for higher privacy levels preserve the more privacy on attributes and attribute values, but cost more computation and communication resources. Based on the security analysis, we show that our scheme satisfies both the verifiability and privacy of attributes and attribute values. According to experimental results, the efficiency of different privacy levels is acceptable for laptops.

Acknowledgement

This work was partially supported by the U.S. National Science Foundation under Grants CNS-0916391.

References

[1] D.K. Zismer, J. McCullough, and P.E. Person, "Integrated health care economics. part 2: Understanding the revenue drivers in fully integrated community health systems.," *Physician Exec*, vol. 35, no. 4, pp. 26–8, 2011.

[2] M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings," *SECURECOMM'10*, pp. 89–106, 2010.

[3] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," *Proceedings of the 2009 ACM workshop on Cloud computing security, CCSW '09*, pp. 103–114, 2009.

[4] J. Jin, G.-J. Ahn, H. Hu, M. J. Covington, and X. Zhang, "Patient-centric authorization framework for sharing electronic health records," *Proceedings of the 14th ACM symposium on Access control models and technologies*, pp. 125–134, 2009.

[5] M. Li, N. Cao, S. Yu, and W. Lou, "Findu: Privacy-preserving personal profile matching in mobile social networks," *INFOCOM 2011. The 30th Conference on Computer Communications. IEEE*, pp. 2435–2443, Apr. 2011.

[6] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure friend discovery in mobile social networks," *INFOCOM 2011. The 30th Conference on Computer Communications. IEEE*, pp. 1647–1655, Apr. 2011.

[7] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," *Proceedings of the 2003 ACM SIGMOD International Conference on Management of data*, pp. 86–97, 2003.

[8] R. Fagin, M. Naor, and P. Winkler, "Comparing information without leaking it," *Commun. ACM*, vol. 39, pp. 77–85, May 1996.

[9] L. Kissner and D. Song, "Private and threshold set-intersection," *Proceedings of CRYPTO '05*, August 2005.

[10] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," *IEEE Pervasive Computing*, vol. 4, pp. 28–34, April 2005.

[11] J. Manweiler, R. Scudellari, and L. P. Cox, "Smile: encounter-based trust for mobile social services," *Proceedings of the 16th ACM conference on Computer and communications security, CCS '09*, pp. 246–255, 2009.

[12] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," *Proceedings of the twentieth annual ACM symposium on Theory of computing STOC'88*, pp. 103–112, 1988.

[13] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," *Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology,EUROCRYPT'08*, pp. 415–432, 2008.

[14] J. Bethencourt, E. Shi, and D. Song, "Signatures of reputation: Towards trust without identity," *In 14th International Conference on Financial Cryptography and Data Security (FC '10)*, January 2010.

[15] J. Groth, "Fully anonymous group signatures without random oracles," *Proceedings of the Advances in Crypotology 13th international conference on Theory and application of cryptology and information security ASIACRYPT'07*, pp. 164–180, 2007.

[16] M. Belenkiy, M.Chase, M. Kohlweiss, and A. Lysyanskaya, "Non-interactive anonymous credentials," *Cryptology ePrint Archive, Report 2007/384*, 2007, http://eprint.iacr.org/.

[17] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[18] M. Barua, X. Liang, R. Lu, and X. Shen, "Peace: An efficient and secure patient-centric access control scheme for ehealth care system," *Computer Communications Workshops, 2011 IEEE Conference on*, pp. 970 –975, april 2011.

[19] S. Narayan, M. Gagné, and R. Safavi-Naini, "Privacy preserving ehr system using attribute-based infrastructure," *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, pp. 47–52, 2010.

[20] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *Advances in Cryptology —CRYPTO 2001*, pp. 213–229, 2001.

[21] C. Shirky, *Here Comes Everybody: The Power of Organizing Without Organizations*, The Penguin Press HC, 1st edition, 2008.

[22] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," *CRYPTO'04, LNCS*, pp. 41–55, 2004.

[23] D. Boneh and X. Boyen, "Short signatures without random oracles and the sdh assumption in bilinear groups," *Journal of Cryptology*, vol. 21, no. 2, pp. 149–177, 2008.

[24] F. Hess, "Efficient identity based signature schemes based on pairings," *Selected Areas in Cryptography*, pp. 310–324, 2003.

[25] P. MacKenzie, M. K. Reiter, and K. Yang, "Alternatives to non-malleability: Definitions, constructions, and applications (extended abstract)," *In proceedings of TCC '04, LNCS*, pp. 171–190, 2004.