On the Security of "Verifiable Privacy-Preserving Monitoring for Cloud-Assisted mHealth Systems"

Hardik Gajera, Shruti Naik, and Manik Lal Das(✉)

DA-IICT, Gandhinagar, India
kidrah123@gmail.com, naik.shruti@outlook.com, maniklal_das@daiict.ac.in

Abstract. Protecting user data in public server is one of the major concerns in cloud computing scenarios. In recent trends, data owner prefers storing data in a third party server in a controlled manner, sometimes in an encrypted form. In this paper, we discuss a recent scheme [1] appeared in INFOCOM 2015 that claims verifiable privacy-preserving service in healthcare systems. We show that the scheme [1] suffers from security weaknesses, in particular, it does not provide privacy-preserving services, which is the main claim of the scheme. We provide an improved solution by slightly modifying the scheme, which retains the security and privacy claim intact without increasing any overhead.

Keywords: Privacy · Cloud security · Access control · Data encryption · Authentication

1 Introduction

Cloud computing is a cost effective computing paradigm for convenient, on-demand data access to a shared pool of configurable computing resources such as networks, servers, storage, applications and services [2]. Broadly, there are three types of consumers in cloud computing – Cloud server as a consumer, Merchant (Data Owner) as a consumer, and Customer as a consumer. Cloud server facilitates storage and services in which merchant stores the application data and all eligible customers of the merchant get on-demand services from the cloud infrastructure. Data owner hires the cloud infrastructure for storing application data in the cloud storage. While resource outsourcing provides significant advantages to data owners as well as to service consumers, there are some important concerns such as security, privacy, ownership and trust that have been discussed substantially over past decade [3–6]. For example, the company can delegate the health monitoring systems to the cloud, where a patient can directly communicate with the cloud. However, upon receiving the patient request the cloud can generate a fabricated report for some malicious intent. Therefore, there is a possibility that cloud server can manipulate the data without data owner's knowledge. In order to avoid such scenarios, data owner can prefers to

I. Ray et al. (Eds.): ICISS 2016, LNCS 10063, pp. 324–335, 2016.
DOI: 10.1007/978-3-319-49806-5_17

store data in cloud server in a controlled manner so that the cloud server cannot manipulate the data while consumer getting services from it. In recent times, several mHealth services have been proposed [4,7–10]. MediNet [7] discussed a mobile healthcare system that can personalize the self-care process for patients with both diabetes and cardiovascular disease. MediNet uses a reasoning engine to make recommendations to a patient based on current and previous readings from monitoring devices connected to the patient and on information that is known about the patient. HealthKiosk [8] proposed a family-based healthcare system that considers contextual information and alerting mechanisms for continuous monitoring of health conditions, where the system design of HealthKiosk has an important entity known as *sensor proxy* that acts as a bridge between the raw data sensed from the sensing device and the kiosk controller, and also acts as a data processing unit. In [9], a taxonomy of the strategies and types of health interventions have been discussed and implemented with mobile phones. Lin *et al.* [4] proposed a cloud-assisted privacy preserving mobile health monitoring system to protect the privacy of users. Their scheme uses the key private proxy re-encryption technique by which the computational cost of the users is primarily done in the cloud server. A basic model for mobile healthcare system is depicted in Fig. 1.

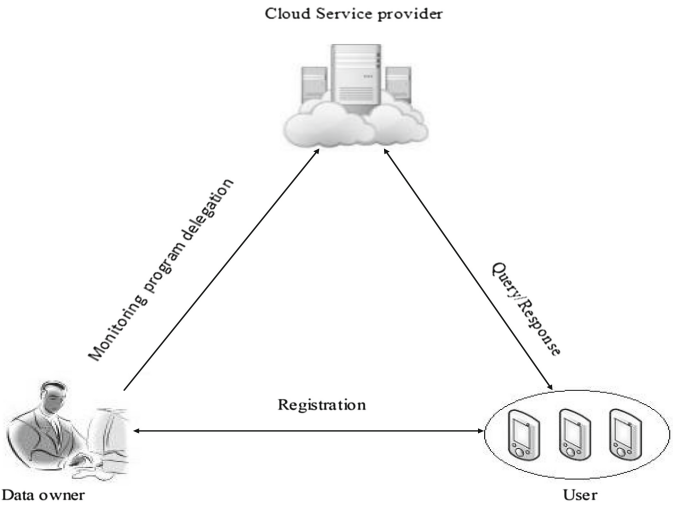

Fig. 1. A basic model for mobile healthcare system

In 2015, Guo *et al.* [1] proposed a scheme for verifiable privacy-preserving monitoring for cloud-assisted health systems. In this paper, we show that the scheme [1] suffers from major security weaknesses, in particular, the scheme does not provide privacy-preserving services, which is the main claim of the scheme. We provide a mitigation for the weaknesses by modifying the scheme. The improved

scheme retains the security and privacy claims of [1] without increasing any overhead.
The remainder of the paper is organized as follows. Section 2 reviews the Guo *et al.*'s scheme. Section 3 shows the security weaknesses of the scheme. Section 4 provides the proposed improvements. We conclude the paper in Sect. 5.

2 Guo *et al.*'s Scheme

Guo *et al.* [1] proposed a scheme, appeared in INFOCOM 2015, that claims verifiable privacy-preserving service in healthcare systems. The scheme has two main objectives - (i) privacy-preserving identity verification, and (ii) verifiable PHR computation. The former provides secure identity verification on cloud without revealing identity of user while later guarantees the correctness of generated PHR. The scheme consists of four entities as follows.

- Trust Authority (TA): TA performs issuance and distributing secret and public parameters to other entities of the scheme.
- Cloud Service Provider (CSP): CSP verifies user identity and computes health record computation using the monitoring program $f(\boldsymbol{x})$ provided by the company.
- Company: Company provides health record computation to users with the help of CSP.
- Users: Users are the consumers for their health services/records.

The scheme works as follows. A user receives a private certificate σ from TA. After receiving σ user asks for a blind signature ψ on σ from the company. After that the user is a registered entity for the monitoring program $f(\boldsymbol{x})$ and the blind signature ψ is issued for the user. Here, $f(\boldsymbol{x})$ is a confidential polynomial function and $\boldsymbol{x}$ is the user's data generated by the user as $\boldsymbol{x} = (x_1, x_2, x_3, \cdots, x_N)$, $x_i \in Z_n^*$. To access the health records the user encrypts the vector and then sends an encrypted vector with ψ to the CSP. User computes $\boldsymbol{c} = \mathrm{E}(\boldsymbol{m})$, where $\boldsymbol{m}$ is monitored raw data and $\mathrm{E}(\cdot)$ is a secure encryption scheme. User then generates proofs on σ which are used for authentication. If public verification of given ψ is done by the CSP then it computes $f(\boldsymbol{x})$ on given $\boldsymbol{c}$. After that the CSP computes the monitoring function and gives results $f(E(\boldsymbol{m}))$ and signature δ to the user. User now decrypts using his secret key and checks for correctness of $f(E(\boldsymbol{m}))$ and δ based on monitored data $\boldsymbol{m}$. The detailed construction of the scheme works with the following phases.

2.1 System Setup

TA sets up the system by choosing the security parameters and the corresponding public parameters.

1. General Setup: TA chooses a security parameter ξ and generates public parameters $\texttt{param} = (n, G, G_1, e)$, where $n = pq$ is the order of group G, p and q are large primes, and e is a bilinear pairing mapping.

2. Partially Blind Signature Setup: TA issues domain public parameter $(g,\ g^s) \in G^2$, where s is a master secret key. TA selects two hash functions $H : \{0,1\}^* \rightarrow G$ and $H_0 : \{0,1\}^* \rightarrow Z_n^*$. A signing key pair $(pk,\ sk)$, where $pk = H(id_c) \in G$ and $sk = H(id_c)^s$ is generated by TA for the company.
3. Monitoring System Setup: TA chooses $g_0 \in G$ and publishes h, where $h = g_0^p \in_R G_q$. TA issues σ after providing ID id_A for user, where $\sigma = g^{\frac{1}{s+id_A}}$. TA gives the private key $sk = q$ to the user.

2.2 Privacy-Preserving Identity Verification

This phase is composed of the following four sub-protocols.

1. **Signature Request**
   $(\theta, \phi) \leftarrow \text{Request}(g, pk, id_A, w)$: User asks for some parameters to company for partially blind signature ψ on σ. Before the request is sent, user and company agree on string $l \in \{0,1\}^n$. Then, the company selects $t \in_R Z_n^*$, calculates $\theta = g^t$, $\phi = H(id_c)^t$ and sends (θ, ϕ) to the user.

2. **Partially Blind Signature Generation Process**
   $\epsilon' \leftarrow \text{BlindSign}(\theta, g^s, \phi, l, \sigma)$: User randomly chooses $\alpha, \beta, \gamma \in_R Z_n^\star$ and calculates $\theta' = \theta^\alpha \cdot \left(g^s\right)^\gamma = g^{\alpha t + \gamma s}$, $\phi' = H(id_c)^{\alpha(\beta+t)} H(l)^{-\gamma}$ and $u = \alpha^{-1} H_0(\sigma \parallel \phi') + \beta$, and sends these to the company. Then, the company calculates

   $$\epsilon = H(id_c)^{s(t+u)} H(l)^t$$

   and sends it back to the user, who unblinds ϵ by calculating $\epsilon' = \epsilon^\alpha$.

3. **Commitment and Proof Generation Process**
   $(com_i, \pi) \leftarrow \text{ProveGen}(\theta', \phi', \epsilon', \sigma, l)$. CSP verifies user's identity by using the blind signature $\psi = (\theta', \phi', \epsilon', \sigma, w)$ as follows.

   $$e(e', g) e(X, \sigma) e(Y, g^{-s}) e(H(l)^{-1}, \theta') \stackrel{?}{=} e(g, g)$$

   where $X = g^{id_A} g^s$ and $Y = \phi' \cdot H(id_c)^{H_0(\sigma \| \phi')}$.
   Note that the verification of the above equation requires the identity id_A of the user along with the blind signature ψ. Therefore, if the user directly sends the blind signature ψ to the CSP, then it reveals the correlation of id_A and the partially blind signature ϵ'.
   Now user generates proofs for the signature and the certificate. For generation of commitments, user chooses $\mu_i, \nu_i \in_R Z_n, i = 1, 2, 3, 4$.

   $\texttt{com}_1 = \epsilon' h^{\mu_1} = H(id_c)^{\alpha s(t+u)} H(l)^{\alpha t} h^{\mu_1}, \texttt{com}'_1 = g h^{\nu_1}$
   $\texttt{com}_2 = g^{id_A + s} h^{\mu_2}, \texttt{com}'_2 = \sigma h^{\nu_2} = g^{\frac{1}{s+id_A}} h^{\nu_2}$
   $\texttt{com}_3 = \phi' \cdot H(id_c)^{H_0(\sigma \| \phi')} h^{\mu_3}, \texttt{com}'_3 = g^{-s} h^{\nu_3}$
   $\texttt{com}_4 = H(l)^{-1} h^{\mu_4}, \texttt{com}'_4 = \theta' h^{\nu_4} = g^{\alpha t + \gamma s} h^{\nu_4}$
   After calculating commitment set, user builds the proof

   $$\pi = \Pi_1^4 (\texttt{com}_i h^{-\mu_i})^{\nu_i} (\texttt{com}'_i)^{\mu_i}$$

   and then sends $(\{\texttt{com}_i, \texttt{com}'_i\}_{i=1}^4, \pi)$ to the CSP for verification.

4. **Identity Verification Process**
   $(0,1) \leftarrow \text{Verify}(\{\texttt{com}_i, \texttt{com}'_i\}_{i=1}^4, \pi, h, e(g,g))$. CSP checks the following equality and returns 1 for successful verification, 0 for unsuccessful verification.

$$\prod_{i=1}^{4} e(\texttt{com}_i, \texttt{com}'_i) = e(g,g)e(\pi,h)$$

2.3 Verifiable PHR Computation

After identity verification, user uploads PHR by the following steps.

1. Monitoring Program Delegation: The company delegates the monitoring program to the cloud and then user's PHR is computed by the cloud. The company sends the coefficient vector $\boldsymbol{a} = (a_0, a_1, \cdots, a_k)$ and string l to the cloud, where l is used for identifying correlation program.
2. PHR Encryption: Let PHR m be an entry from data vector $\boldsymbol{m} = (m_1, m_2, \cdots, m_N), m_i \in Z_n$. User chooses a set of random numbers $\boldsymbol{r} = (r_0, r_1, \cdots, r_k), r_i \in Z_n$. Then, the user sends $\boldsymbol{r}$ to the company. After getting $\boldsymbol{r}$, the company calculates $\boldsymbol{r'} = \boldsymbol{r} \cdot \boldsymbol{a} = (a_0 r_0, a_1 r_1, \cdots, a_k r_k)$. Then, company sends $h^{\bar{r}} = h^{\sum_{i=0}^{k} r'_i}$ and $g^{\bar{r}}$ to the user, and $\bar{r}$ to the CSP, where $\bar{r} = \sum_{i=0}^{k} a_i r_i$. User picks $d \in_R Z_n$ and generates the ciphertext of PHR as

$$c = \left(g h^{d \cdot r_0}, g^m h^{d \cdot r_1}, g^{m^2} h^{d \cdot r_2}, \cdots, g^{m^k} h^{d \cdot r_k}\right)$$

   where each entry is computed as $c_i = g^{m^i} \cdot (h^{r_i})^d$. Now, user sends $\{c, \lambda, H(l)\}$ to the CSP, where $\lambda = \frac{1}{(x-m)\cdot d} \bmod n$. User also requests the company to compute a public parameter $g^{f(x)}$, which later the company sends to the CSP.
3. Verifiable PHR Computation: PHR is computed as follows.
   $\upsilon = \prod_{i=0}^{k} \left(g^{m^i} \cdot (h^{r_i})^d\right)^{-a_i} = \prod_{i=0}^{k} g^{-a_i \cdot m^i} \cdot h^{-a_i r_i d} = g^{\sum_{i=0}^{k} -a_i \cdot m^i} \cdot h^{\sum_{i=0}^{k} -a_i r_i d}$
   $= g^{-f(m)} \cdot h^{-d \sum_{i=0}^{k} r'_i}$

   CSP computes $\lambda' = \frac{\lambda}{\bar{r}} = \frac{1}{(x-m)\cdot d \cdot \bar{r}}$ and signature δ using $g^{f(x)}$ as,
   $\delta = \left(g^{f(x)} \cdot \upsilon\right)^{\lambda'} = \left(g^{f(x)-f(m)} \cdot h^{-d \sum_{i=0}^{k} r'_i}\right)^{\frac{1}{(x-m)\cdot d \cdot \bar{r}}}$
   $= g^{\frac{f(x)-f(m)}{(x-m)} \cdot \frac{1}{d\bar{r}}} \cdot h^{-\frac{1}{(x-m)}} = \left(g^{w(x)} \cdot h^{-\frac{d\bar{r}}{(x-m)}}\right)^{\frac{1}{d\bar{r}}}$
   where $w(x)$ is a $(k-1)$-degree polynomial function. If $f(m)$ is the value based on m, then only it satisfies this condition $w(x) \equiv \frac{f(x)-f(m)}{(x-m)}$. Then, CSP sends $\{\upsilon, \delta\}$ to the user.
4. PHR Result Decryption and Verification: Using the private key $sk = q$ the user decrypts υ as

$$\left(\frac{1}{\upsilon}\right)^q = \left(g^{f(m)} h^{d\bar{r}}\right)^q = \left(g^q\right)^{f(m)} h^{d\bar{r}q} = \left(g^q\right)^{f(m)} \in G_p.$$

User can recover $f(m)$ by computing the discrete log of $\left(\frac{1}{\upsilon}\right)^q$ with base g^q. Here, $f(m)$ is bounded by M where M is very small compared to p,q and therefore, it is feasible to compute the discrete log of $\left(\frac{1}{\upsilon}\right)^q$.

For getting the proof on $f(m)$, the user sends encrypted $(x, f(m))$ to the company. Then, the company constructs coefficient vector $w(x)$ as $\boldsymbol{w} = (w_0, w_1, \cdots, w_{k-1})$ and proves $W = g^Z$, where $Z = \sum_{i=0}^{k-1} w_i x^i$ and responds to the user. Now, the user calculates $(g^{\bar{r}})^d = g^{d\bar{r}}$ and $\eta = (h^{\bar{r}})^{-d/(x-m)}$. Finally, the user verifies the following equation to see whether the CSP has computed correct results or not.

$$e(W \cdot \eta, g) \stackrel{?}{=} e(\theta, g^{d\bar{r}}).$$

3 Security Weaknesses in Guo *et al.*'s scheme

We show two security flaws in Guo *et al.*'s scheme [1]. The company's goal is the confidentiality of the monitoring program $f(x)$. If a malicious user obtain $f(x)$ then he can use it for free and he can even sell it to someone else. We note that the company delegates the monitoring program $f(x)$ to the CSP, with the assumption that the computation of $f(x)$ on patients' PHR can be done by the CSP without loosing the confidentiality of the monitoring program $f(x)$. In other words, the monitoring program $f(x)$ should not be known to any other party except the Company and the CSP. Furthermore, anyone can pass the identity verification process without even communicating with the TA or company and therefore, if a malicious user leaks $H(l)$ to a non-user (attacker), then the attacker can use the system with all credentials.

3.1 Insider Attack

The monitoring program is a polynomial of degree k and hence, it can be represented as a $k+1$ length vector, $\boldsymbol{a} = (a_0, a_1, a_2, \ldots, a_k)$, where a_i is the coefficient of x^i in the polynomial.

$$f(x) = \sum_{i=0}^{k} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k.$$

The company wants to keep this vector $\boldsymbol{a}$ secret from everyone except the cloud. Therefore, there are total $k+1$ unknowns and it is easy to find values for these unknowns if we have $k+1$ linearly independent equations involving the coefficients $\{a_i\}_{i=0}^k$. An authenticated user(insider) can use the service for $k+1$ times and get PHR report $f(m_i)$, where m_i is the PHR sent by the user on i^{th} time use of the service. Using the set $\{(m_i, f(m_i))\}_{i=0}^k$, the user can create the system of equations in $k+1$ variable and solve it for the vector $\boldsymbol{a}$. More concretely, assume

that the user has the set $\{(m_i, f(m_i))\}_{i=0}^{k}$. Then for each $i \in \{0, 1, 2, \ldots, k\}$, we have

$$a_0 + a_1 m_i + a_2 m_i^2 + \cdots + a_k m_i^k = f(m_i)$$

Without loss of generality, we assume that these $(k+1)$ equations are linearly independent (if not, then the user can always use the service until it is true). We can represent this system of equation in terms of matrices as follows.

$$A = \begin{bmatrix} 1 & m_1 & \ldots & m_1^k \\ 1 & m_2 & \ldots & m_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & m_{k+1} & \ldots & m_{k+1}^k \end{bmatrix} \quad X = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} \quad B = \begin{bmatrix} f(m_1) \\ f(m_2) \\ \vdots \\ f(m_k) \end{bmatrix}$$

$$AX = B$$

Solution of the above system of equations is given by

$$X = A^{-1}B.$$

Now, the user can easily solve the above system of equation for the vector $X = (a_0, a_1, \ldots, a_k)$ and the user can use it to compute $f(m) = \sum_{i=0}^{k} a_i m^i$ for any PHR m. In the scheme [1], it is assumed that the degree of the polynomial is around 10 and that makes this attack more easy. Although this attack does not violate privacy of other users, it reveals the confidential monitoring program $f(x)$ of all users pertaining to the company who owns the monitoring program. In this attack, the user obtains $f(x)$ and thereby, computes the result of $f(x)$ without contacting the CSP or the Company, which reveals the confidentiality of the monitoring program $f(x)$.

3.2 Outsider Attack

We note that the cloud does not use any extra information other than the commitments sent by the user and the public parameters published by TA. This makes the process vulnerable to unauthenticated identity verification. The attacker can choose commitments as follows.

$\mathtt{com}_1 = g,\ \mathtt{com}'_1 = g$
$\mathtt{com}_2 = g,\ \mathtt{com}'_2 = g^{-2}$
$\mathtt{com}_3 = g,\ \mathtt{com}'_3 = g^2$
$\mathtt{com}_4 = \pi,\ \mathtt{com}'_4 = h$, where $\pi \in_R G$

Since g and h are public parameters, the attacker does not have any trouble in choosing these commitments and π can be any random element of the group G. The attacker sends π and $(\{\mathtt{com}_i, \mathtt{com}'_i\}_{i=1}^{4})$ to the cloud for verification. Upon receiving the commitments from the user, the cloud verifies the equality of the following equation.

$$\prod_{i=1}^{4} e(\mathtt{com}_i, \mathtt{com}'_i) = e(g, g)e(\pi, h)$$

Proof. We prove the equality of the above equation.

$$\begin{aligned}
&\textstyle\prod_{i=1}^{4} e(\mathtt{com}_i, \mathtt{com}'_i) \\
&= e(g,g)e(g,g^{-2})e(g,g^2)e(\pi,h) \\
&= e(g,g)e(g,g)^{-2}e(g,g)^2e(\pi,h) \\
&= e(g,g)^{1-2+2}e(\pi,h) \\
&= e(g,g)e(\pi,h)
\end{aligned}$$

Therefore, anyone can pass through the identity verification process. Once the verification is successful, the cloud allows the attacker to use the service. Here, we assume that the attacker already has $H(l)$ and k. The attacker follows the rest of the process same as an authenticated user described in the previous section and gets (υ, δ) in response from the cloud, where

$$\upsilon = g^{-f(m)}h^{-d\overline{r}}$$

The υ contains information about $f(m)$ and the attacker's aim is to get the PHR report $f(m)$ for the PHR m. Note that the attacker is not an authenticated user and he does not have the secret key q and hence, can not decrypt υ. However, the attacker can find $f(m)$ using brute force because size of $f(m)$ is small. Since the attacker follows the rest of the process after identity verification, the attacker will have d and $h^{\overline{r}}$. The attacker computes

$$\upsilon' = \upsilon h^{d\overline{r}} = g^{-f(m)}.$$

In [1], the authors have considered that values of m and $f(m)$ are not more than 1000. Therefore, the attacker can simply check whether υ' is equal to g^{-j} for every $j \in \{0, 1, 2, \ldots, 1000\}$. By using only 1000 iterations, the attacker can successfully get $f(m)$.

4 Proposed Improvements

4.1 Prevention of Insider Attack

The insider attack is possible because the attacker knows the degree k of the polynomial. We provide a way to keep the polynomial $f(x)$ secure by keeping the degree of the polynomial secret.

Let m be the PHR value and user wants to get a report $f(m)$ for it. User chooses two random numbers $r_0, d \in Z_n$ and a random prime p_1. User computes $m' = m + p_1$ and sends (r_0, d, m') to the company. The Fig. 2 reflects the changes suggested for preventing the observed insider attack in Guo *et al.*'s scheme.

After receiving (r_0, d, m'), the company generates k random integers $r_1, r_2, \ldots, r_k \in Z_n$ using r_0. Company calculates

$$\boldsymbol{r'} = \boldsymbol{r}.\boldsymbol{a} = (a_0 r_0, a_1 r_1, \ldots, a_k r_k)$$

and

$$c = (gh^{dr_0}, g^{m'}h^{dr_1}, g^{m'^2}h^{dr_2}, \ldots, g^{m'^k}h^{dr_k}).$$

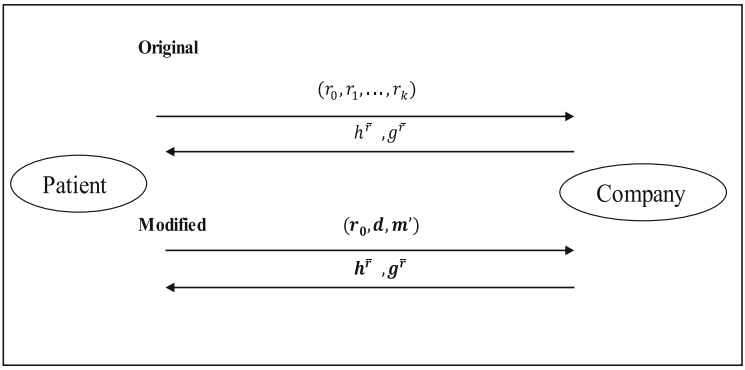

Fig. 2. Prevention of insider attack: changes between Original and Modified schemes

Company sends $(h^{\bar{r}}, g^{\bar{r}})$ to the user and $(\bar{r}, c)$ to the cloud, where $\bar{r} = \sum_{i=0}^{k} a_i r_i$. User selects a random point $x \in Z_n$ and computes $\lambda = \frac{1}{x-m'}d$. User sends x to the company and $(\lambda, H(l))$ to the cloud. Company computes $g^{f(x)}$ and sends it to the cloud. Upon receiving $(\lambda, H(l))$ from user and $(\bar{r}, c, g^{f(x)})$ from the company, the cloud computes υ and δ. Everything remains same except that c is encryption of m' instead of m. After decrypting υ, user gets $f(m') = f(m + p_1)$. For sufficiently large value of p_1, we have $f(m + p_1) \bmod p_1 = f(m)$. The verification process remains same. Since the user does not know the degree k, the user can not retrieve coefficients of the polynomial $f(x)$.

4.2 Prevention of Outsider Attack

We modify the scheme in such a way that only registered user can use the service to get PHR report $f(m)$ for a given PHR m. Note that the cloud computes $f(m)$ only after successful identity verification process. After generation of the blind signature, the company and the cloud agree on some random number $z \in_R Z_{p^*}$ and a timestamp t_m. Then, the company computes $g_1 = g^{H(t_m \| z)}$ and sends g_1 with ϵ to the user. The Figs. 3 and 4 reflect the changes suggested for preventing the observed outsider attack in Guo *et al.*'s scheme.

After receiving $\{g_1, \epsilon\}$ the user computes commitments. Except $\mathtt{com}_2$ all other commitments remain same. We modify $\mathtt{com}_2$ as follows:

$$\mathtt{com}_2 = g_1^{id_A + s} h^{\mu_2}$$

Now, based on this modification, user computes the proof

$$\pi = \prod_{i=1}^{4} (\mathtt{com}_i h^{-\mu_i})^{\nu_i} (\mathtt{com}'_i)^{\mu_i}$$

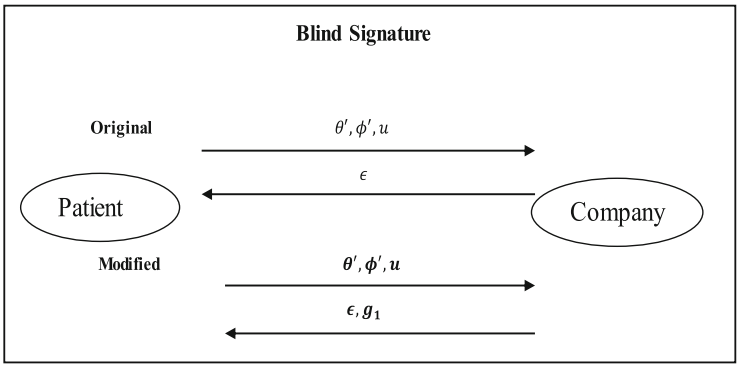

Fig. 3. Prevention of outsider attack: changes shown in Blind signature

and sends $(\{\mathtt{com}_i, \mathtt{com}'_i\}_{i=1}^4, \pi)$ to the cloud for verification. During the identity verification process, the cloud verifies the equality of following equation and returns 1 for successful verification and 0 for unsuccessful verification.

$$\prod_{i=1}^{4} e(\mathtt{com}_i, \mathtt{com}'_i) = e(g_1, g)e(\pi, h).$$

Correctness:

$$\prod_{i=1}^{4} e(\mathtt{com}_i, \mathtt{com}'_i)$$

$$= e(\epsilon' h^{\mu_1}, gh^{\nu_1})e(\phi' \cdot H(id_c)^{H_0(\sigma\|\phi')}h^{\mu_3}, g^{-s}h^{\nu_3})$$

$$\cdot e(H(l)^{-1}h^{\mu_4}, g^{\alpha t+\gamma s}h^{\nu_4})e(g_1^{id_A+s}h^{\mu_2}, g^{\frac{1}{s+id_A}}h^{\nu_2})$$

$$= e(H(id_c)^{\alpha s(t+u)}H(l)^{\alpha t}, g)e(\phi' \cdot H(id_c)^{H_0(\sigma\|\phi')}, g^{-s})$$

$$\cdot e(H(l)^{-1}, g^{\alpha t+\gamma s})e(h^{\mu_1}, g)e(\epsilon' h^{\mu_1}, h^{\nu_1})e(h^{\mu_3}, g^{-s})$$

$$\cdot e(\phi' \cdot H(id_c)^{H_0(\sigma\|\phi')}h^{\mu_3}, h^{\nu_3})e(h^{\mu_4}, g^{\alpha t+\gamma s})$$

$$\cdot e(H(l)^{-1}h^{\mu_4}), h^{\nu_4})e(g_1^{id_A+s}h^{\mu_2}, g^{\frac{1}{s+id_A}}h^{\nu_2})$$

$$= e(H(id_c)^{\alpha s(t+u)}, g)e(H(id_c)^{\alpha(\beta+t)+H_0(\sigma\|\phi')}, g^{-s})$$

$$\cdot e(H(l)^{\alpha t}, g)e(H(l), g)^{\gamma s-\alpha t-\gamma s}e(h^{\mu_1}, g)e((\epsilon' h^{\mu_1})^{\nu_1}, h)$$

$$\cdot e(g^{-s\mu_3}, h)e((\phi' \cdot H(id_c)^{H_0(\sigma\|\phi')}h^{\mu_3})^{\nu_3}, h)e(g^{\mu_4(\alpha t+\gamma s)}, h)$$

$$\cdot e(H(l)^{-1}(h^{\mu_4})^{\nu_4}, h)e(g_1, g)e(g_1^{(id_A+s)\nu_2+\frac{\mu_2}{s+id_A}}, h)e(h^{\mu_2\nu_2}, h)$$

$$= e(g_1, g)\prod_{i=1}^{4} e((com_i h^{-\mu_i})^{\nu_i}(com_i')^{\mu_i}, h) = e(g_1, g)e(\pi, h)$$

Here, the attacker does not have g_1, so he can not pass the identity verification process. Without passing the verification process, the attacker can not compute $f(m)$ for any PHR m. We note that after the identity verification there is also a need for message authentication (to avoid user impersonation attack) between the company and the user in the PHR computation phase of the scheme.

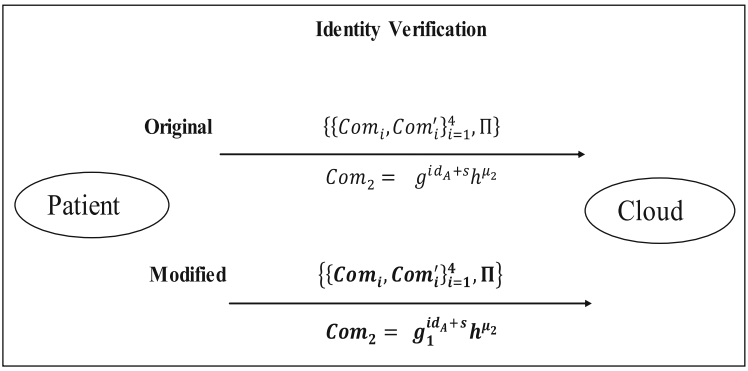

Fig. 4. Prevention of outsider attack: changes shown for Identity Verification

4.3 Performance Analysis

We compare the Guo *et al.*'s scheme and the proposed improved scheme with respect to the computational, storage and communication costs requirement in the schemes. In the Table 1, k is the degree of the monitoring program $f(x)$, n is a public parameter, and M is the size of the message space. The Table 1 provides computational complexity of both schemes in terms of the number of group multiplications (G), the number of integer multiplications (M) and the number of bilinear pairing computations (E). For exponentiation of a group element, we consider the square and multiply algorithm to count the number of group multiplications. The improved scheme is comparable with Guo *et al.*'s

Table 1. Comparison of Guo *et al.*'s scheme and the improved scheme

	Guo *et al.*'s scheme [1]	Improved scheme
Computational cost	$((3k + M + 40)log(n) + 2k + 25)G + 6(k + 1)M + 8E$	$((3k + M + 40)log(n) + k + 25)G + (5k + 7)M + 8E$
Storage cost	*Public:* $7log(n)$ *bits* *Private:* (k+6)$log(n)$ *bits*	*Public:* $7log(n)$ *bits* *Private:* (k+7)$log(n)$ *bits*
Communication cost	(3k+33)$log(n)$ *bits*	(k+35)$log(n)$ *bits*

scheme in terms of computation and storage costs and provides better efficiency in terms of communication cost.

5 Conclusion

We have discussed a recent work on verifiable privacy-preserving service in healthcare systems [1], appeared in INFOCOM 2015. We have shown that the scheme does not provide privacy-preserving services, suffers from insider and outsider attacks. We have suggested for mitigation by modifying the scheme. The improved scheme retains the security and privacy claim without increasing any overhead.

Acknowledgment. This work was supported in part by Indo French Centre for the Promotion of Advanced Research(IFCPAR) and Center Franco-Indien Pour LA Promotion DE LA Recherche Avancee(CEFIPRA) through the project DST/CNRS 2015-03 under DST-INRIA-CNRS Targeted Programme.

References

1. Guo, L., Fang, Y., Li, M., Li, P.: Verifiable privacy-preserving monitoring for cloud-assisted mHealth systems. In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM 2015), pp. 1026–1034 (2015)
2. Shawish, A., Salama, M.: Cloud computing: paradigms and technologies. In: Bessis, N., et al. (eds.) Inter-cooperative Collective Intelligence: Techniques and Applications. Studies in Computational Intelligence, vol. 495, pp. 39–67. Springer, Heidelberg (2014)
3. Guo, L., Zhang, C., Yue, H., Fang, Y.: PSaD: a privacy-preserving social-assisted content dissemination scheme in DTNs. IEEE Trans. Mob. Comput. **13**(12), 2903–2918 (2014)
4. Lin, H., Shao, J., Zhang, C., Fang, Y.: CAM: cloud-assisted privacy preserving mobile health monitoring. IEEE Trans. Inf. Forensics Secur. **8**(6), 985–997 (2013)
5. Basu, A., Sengupta, I., Sing, J.K.: Secured cloud storage scheme using ECC based key management in user hierarchy. In: Jajodia, S., Mazumdar, C. (eds.) ICISS 2011. LNCS, vol. 7093, pp. 175–189. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25560-1_12
6. Wang, W., Li, Z., Owens, R., Bhargava, B.: Secure and efficient access to outsourced Data. In: Proceedings of the ACM Workshop on Cloud Computing Security (CCSW 2009), pp. 55–66 (2009)
7. Mohan, P., Sultan, S.: MediNet: a mobile healthcare management system for the Caribbean region. In: Proceedings of International Conference of Mobile and Ubiquitous Systems: Networking & Services, pp. 1–2 (2009)
8. Liu, C.H., Wen, J., Yu, Q., Yang, B., Wang, W.: Health Kiosk: a family-based connected healthcare system for long-term monitoring. In: Proceedings of IEEE Conference on Computer Communications Workshops, pp. 241–246 (2011)
9. Klasnja, P., Pratt, W.: Healthcare in the pocket: mapping the space of mobile-phone health interventions. J. Biomed. Inform. **45**(1), 184–198 (2012)
10. Chiarini, G., Ray, P., Akter, S., Masella, C., Ganz, A.: mHealth technologies for chronic diseases and elders: a systematic review. IEEE J. Sel. Areas Commun. **31**(9), 6–18 (2013)