Privacy Towards Cloud Based Mobile Health Monitoring Services

S.Manjunath,N.KoteswaraRao

Abstract—Cloud-assisted mobile health (mHealth) monitoring, which applies the prevailing mobile communications and cloud computing technologies to provide feedback decision support, has been considered as a revolutionary approach to improving the quality of healthcare service while lowering the healthcare cost. Unfortunately, it also poses a serious risk on both clients' privacy and intellectual property of monitoring service providers, which could deter the wide adoption of mHealth technology. This paper is to address this important problem and design a cloudassistedprivacy preserving mobile health monitoring system to protect the privacy of the involved parties and their data. Moreover, the outsourcing decryption technique and a newlyproposed key private proxy re-encryption are adapted to shift the computational complexity of the involved parties to the cloud without compromising clients' privacy and service providers' intellectual property. Finally, our security and performance analysis demonstrates the effectiveness of our proposed design.

Key Words—Mobile health (mHealth), Healthcare, Privacy, Outsourcing decryption, Key private proxy re-encryption.

INTRODUCTION

Wide deployment of mobile devices, such as smart phones equipped with low cost sensors, has already shown great potential in improving the quality of healthcare services. Remote mobile health monitoring has already been recognized as not only a potential, but also a successful example of mobile health (mHealth) applications especially for developing countries. The Microsoft launched project "MediNet" is designed to realize remote monitoring on the health status of diabetes and cardiovascular diseases in remote areas in Caribbean countries [1]. In such a remote mHealth monitoring system, a client could deploy portable sensors in wireless body sensor networks to collect various physiological data, such as blood pressure (BP), breathing rate (BR), Electrocardiogram (ECG/EKG), peripheral oxygen saturation (SpO2) and blood glucose. Such physiological data could then be sent to a central server, which could then run various web medical applications on these data to return timely advice to the client. These applications may have various functionalities ranging from sleep pattern analyzers, exercises, physical activity assistants, to cardiac analysis systems, providing various medical consultation [2]. Moreover, as the emerging cloud computing technologies evolve, a viable solution can be sought by incorporating the software as a service (SaaS) model and pay-as-you-go business model in cloud computing, which would allow small companies (healthcare service providers) to excel in this healthcare market. It has been observed that the adoption of automated decision support algorithms in the
cloud-assisted mHealth monitoring has been considered as anb future trend [3]. Although the existing privacy laws such as HIPAA (Health Insurance Portability and Accountability Act) provide baseline protection for personal health record, they are generally considered not applicable or transferable to cloud computing environments [6]. Besides, the current law is more focused on protection against adversarial intrusions while there is little effort on protecting clients from business collecting private information. Meanwhile, many companies have significant
commercial interests in collecting clients' private health data [7] and sharing them with either insurance companies, research institutions or even the government agencies. It has also been indicated [8] that privacy law could not really exert any real protection on clients' data privacy unless there is an effective mechanism to enforce restrictions on the activities of healthcare service providers. As an important remark, our design here mainly focuses on insider attacks, which could be launched by either malicious or non-malicious insiders. For instance, the insiders could be disgruntled employees or healthcare workers who enter the healthcare business for criminal purpose [21], [22]. It was reported that 32% of medical data breaches in medical establishments between January 2007 and June 2009 were due to insider attacks [23], and the incident rate of insider
attacks is rapidly increasing [23]. The insider attacks have cost the victimized institutions much more than what outsider attacks have caused [24].

In this paper, we design a cloud-assisted mHealth monitoring system (CAM). We first identify the design problems on privacy preservation and then provide our solutions. To ease the understanding, we start with the basic scheme so that we can identify the possible privacy breaches. We then provide an improved scheme by addressing the identified privacy problems. The resulting improved scheme allows the mHealth service provider (the company) to be offline after the setup stage and enables it to deliver its data or programs to the cloud securely. To reduce clients' decryption complexity, we incorporate the recently proposed outsourcing decryption technique [25] into the underlying multi-dimensional range queries system to shift clients' computational complexity to the cloud without revealing any information on either clients' query input or the decrypted decision to the cloud. input signal, and finally, the load also helps with common

I.SYSTEM AND ADVERSARIAL MODELS

To facilitate our discussion, we first elaborate our cloudassisted mHealth monitoring system (CAM). CAM consists of four parties: the cloud server (simply the cloud), the company who provides the mHealth monitoring service (i.e., the healthcare service provider), the individual clients (simply clients), and a semi-trusted authority (TA). The company stores its encrypted monitoring data or program in the cloud server. Individual clients collect their medical data and store them in their mobile devices, which then transform the data into attribute vectors. A semi-trusted authority is responsible for distributing private keys to the individual clients and collecting the service fee from the clients according to a certain business model such as pay-as-you-go

business model. We assume a neutral cloud server, which means it neither colludes with the company nor a client to attack the other side. This is a reasonable model since it would be in the best business interest of the cloud not to be biased. We admit that it remains possible for the cloud to collude with other malicious entities in our CAM, and we leave the CAM design under these stronger models as future work. We also do not assume that an individual client colludes with other clients. Our security model does not consider the possible side-channel attack [26], [27] due to the co-residency on shared resources either because it could be mitigated with either system level protection [27] or leakage resilient cryptography [28]. CAM
assumes an honest but curious model, which implies all parties should follow the prescribed actions and cannot be arbitrarily malicious. In the following, we briefly introduce the four major steps of CAM: Setup, Store, TokenGen and Query. We only illustrate the functionality of these components in this section while leaving the details in later sections.

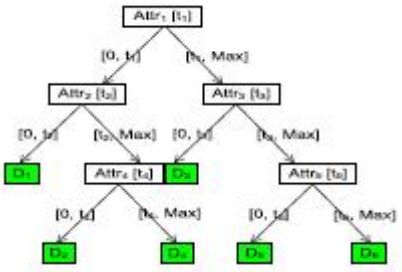

Fig. 1. Branching program

II. SOME PRELIMINARIES AND SECURITY BUILDING BLOCKS

Bilinear Maps

Pairing is crucial to our design, which would further serve as the building blocks of our proposed CAM. A pairing is an efficiently computable, non-degenerate function, $e : G \times G \rightarrow GT$, with the bilinearity property: $e(gr, gs) = e(g, g)rs$ for any $r, s \; Z * q$, the finite field modulo q, where G, and GT are all multiplicative groups of prime order q, generated by g and $e(g, g)$, respectively. It has been demonstrated that the proposed IBE is secure under the decisional bilinear Diffie-Hellman (DBDH) assumption (which states that in the IBE setting, given (g, ga, gb, gc, S), it is computationally difficult to decide whether $S = gabc$). Details can be found in [30].

Branching program

In this section, we formally describe the branching programs [31], which include binary classification or decision trees as a special case. We only consider the binary branching program (as shown in Fig. 1) for the ease of exposition since a private query protocol based on a general decision tree can be easily derived from our scheme. Let v=($v1, ???, vn$) be the vector of clients' attributes. To be more specific, an attribute component vi is a concatenation of an attribute index and the respective attribute value. For instance, A||KW1 might correspond to "blood pressure: 130". Those with a blood pressure lower than 130 are considered as normal, and those above this threshold are considered as high blood pressure.

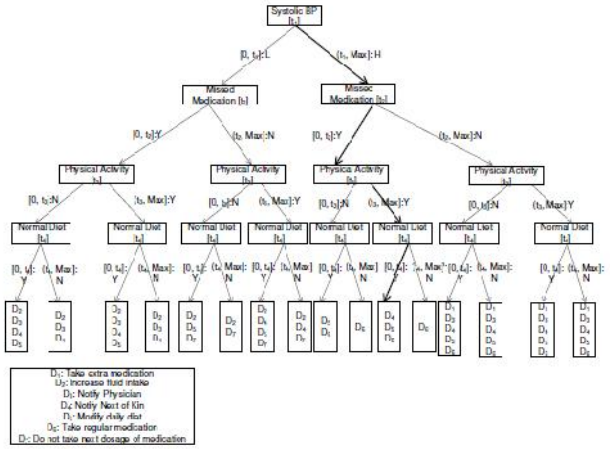

Fig. 2. Using branching program to represent a real monitoring program in MediNet project

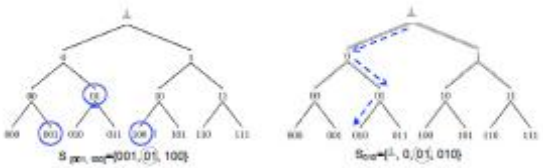

Fig. 3. Basic idea of MDRQ

To illustrate how a practical monitoring program can be transformed into a branching program, we use the monitoring program introduced in the MediNet project [32], [33] to construct a branching program as shown in Fig. 2. The MediNet aims to provide automatic personalized monitoring service for patients with diabetes or cardiovascular diseases. Clients input their related health data such as systolic blood pressure (BP), whether they missed daily medications or had an abnormal diet, and the energy consumption of physical activity to the decision support system, which will then return a recommendation on how the clients can improve their conditions. For instance, assume a hypertension patient inputs an attribute vector consisting of the following elements "[Systolic BP: 150, Missed one medication=0 (indicating he did miss the medication), Energy Expenditure: 900 kcal, salt intake: 1000 milligrams]" and the respective threshold is "$t1$ = 130, $t2$ = 0, $t3$ = 700$kcal$, $t4$ = 1500". The recommendation returned from the monitoring program (Fig. 2) would be "$D4,D5,D6$" (by following the path through comparing each attribute element with the respective thresholdnb at each node), which indicates the clients need to "notify next kin, modify daily diet, and take regular medication".

Homomorphic encryption

Homomorphic encryption is widely used as an underlying tool for constructing secure protocols in the literature [34], [35]. CAM adopts a semantically secure additively homomorphic public-key encryption technique. Intuitively, for homomorphic encryption HEnc(·), given two encrypted messages HEnc($m1$) and HEnc($m2$), the encryption of the addition of the two underlying messages can be computed additively as follows: HEnc($m1$+$m2$) = HEnc($m1$) ⋆HEnc($m2$), where ⋆ is the corresponding operation in the ciphertext space. A typical additively

homomorphic encryption scheme was proposed by Paillier cryptosystem [36], [37]. Homomorphic encryption enables a client to obtain the token corresponding to the input attribute vectors obliviously from TA.

Decryption outsourcing

The pairing-based IBE system [30] and its extensions such as attribute-based encryption [41], [42] has a reputation of costly decryption workload due to the bilinear paring operations in the decryption steps. Moreover, the pairing computation is considered to be especially computationally intensive for resource-constrained mobile phones. For example, for a chosen pairing function, the computation time on a PC with 2.40GHz Intel(R) Core 2 Quad, 3 GB RAM, and Windows 7 is 14.65ms while that on an Android 2.3.2 with 1GHz ARM Cortex A8 and 512 MB RAM is as high as 332.9 ms. Thus, we seek decryption outsourcing to ease the computational complexity. The decryption outsourcing in ABE was first proposed by Green *et al* [25]. It enables a client to transform his secret key to the transformation key and then delegates it to an untrusted server (e.g., a cloud) to use it to transform the original ciphertext into an El Gamal encryption of the original message. The client only needs to compute simple exponentiation operations to obtain the underlying message. In CAM, we intend to apply the outsourcing decryption technique to MDRQs based on the BF-IBE scheme. The BF-IBE based outsourcing decryption is shown as follows. AnonSetup(1_): This algorithm is exactly the same as the original BF-IBE. orrect. We observe that in this construction the client only needs to compute one exponentiation in order to obtain the message, and the costly pairing operation is completed by the cloud. It can be shown as done in [25] that our proposed BFIBE with outsourcing decryption is secure against replayable chosen ciphertext attack (CCA), which implies that the following mask privacy: TA obtains no useful information on the client's identity id since $H1(\mathrm{id})z$ is just a random element to TA under random oracle model.

Key private proxy re-encryption (PRE)

Another technique we will use is the proxy re-encryption (PRE), which was first proposed by Blaze et al. [43], and further formalized by Ateniese et al. [44]. Proxy re-encryption allows an untrusted proxy server with a re-encryption key (rekey) $rkA \rightarrow B$ to transform a ciphertext (also known as first level ciphertext) encrypted for Alice (delegator) into one (second level ciphertext) that could be decrypted by Bob (delegatee) without letting the proxy obtain any useful information on the underlying message. Proxy re-encryption can be categorized according to various properties: unidirectional or bidirectional, non-interactive or interactive, collusion resistant or not, key private or not, and transferable or non-transferable. In our scheme, we emphasize two most relevant properties: unidirectionality nand key privateness. *Unidirectionality* means that delegation from $A \rightarrow B$ does not allow delegation in the opposite direction. *Key privateness* implies that given the rekey $rkA \rightarrow B$, the proxy deduces no information on either the identity of the delegator or the delegatee. In CAM, the monitoring program delivered by the company is encrypted using an MDRQs scheme and the ciphertext is stored in the untrusted cloud. The company then delivers several reencryption keys to the cloud.

III. CAM DESIGN

We are ready to present our design *CAM: cloud-assisted privacy preserving mHealth monitoring system*. To illustrate the fundamental idea behind this design, we start with the basic scheme, and then demonstrate how improvements can be made step-by-step to meet our design goal. Some of the variables in the following illustration may have already been defined in the previous sections. The system time is divided into multiple time periods, called *slots*, each of which can last a week or a month depending on specific application scenarios. There is an estimated maximum number of users N requesting access to the monitoring program in any given slot. When a client attempts to access the program, it is assigned an index [1,N] by TA.

Basic CAM

The following basic scheme runs the BF-IBE system asm a sub-routine and is the fundamental building block in our overall design.

Setup: This algorithm is performed by TA, which publishes the system parameters for the BF-IBE scheme. Store: This algorithm is performed by the company. For each node pj whose child nodes are not leaf nodes, the company runs $CL(j)$ = AnonEnc(id, $PP,L(j)$) and $CR(j)$ = AnonEnc(id, $PP,R(j)$) to encrypt the child node indices under id with either id $S[0;tj]$ or id $S[tj+1;Max]$, respectively. When the child nodes of pj are leaf nodes, the company generates the ciphertext as $CL(j)$ = AnonEnc(id, $PP,mL(j)$) and $CR(j)$ = AnonEnc(id, $PP,mR(j)$), where $mL(j)$ and $mR(j)$ denote the attached information at the two leaf nodes, respectively.

CAM with Full Privacy Preservation

The basic scheme has the following security weakness: first, the identity representation set for a client's attribute vector v is known to TA, and hence TA can easily infer all the client's private attribute vector. Second, the client cannot bprotect his privacy from the cloud either because the cloud can easily find out the identity representation for the private key skvi , i [1, n] by running identity test in MDRQs. The cloud can simply encrypt a random message under any attribute value v' until when it can use skvi to successfully decrypt the ciphertext, which means there is a match between $v' = vi$ and hence it successfully finds out vi. Third, neither can the data privacy of the company be guaranteed since the identity representation of the respective range is revealed to the cloud whenever the decryption is successful due to the match revealing property (see Sec. III-D) of MDRQs. The cloud can finally figure out most of the company's branching program since it has the private keys of all the system users. The improvement consists of four steps just as in the basic scheme.We will show how this improvement meets the desired security requirements.

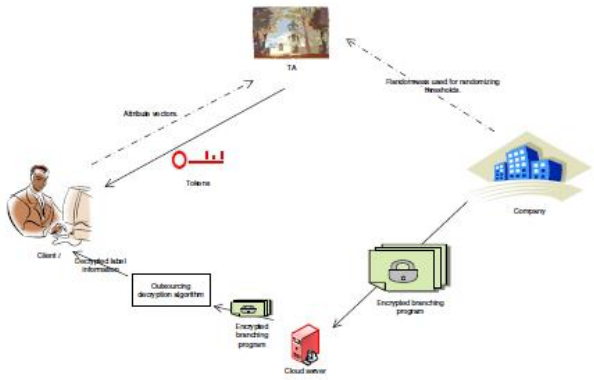

Fig. 4. CAM with Full Privacy Preservation

Although the above improved scheme does meet the desired security requirements, the company may need to compute all the ciphertexts for each of N clients, which implies huge computational overheads and may not be economically feasible for small mHealth companies. In this section, we provide a further improvement to reduce both the computational burden on the company and the communication overhead for the cloud. The high level idea (as shown in Fig. 5) is as follows. We employ a newly developed key private re-encryption scheme (introduced in Sec. IV-C1) as an underlying tool. Instead of computing a ciphertext for each client, the company generates one single ciphertext, which will then be delivered to the cloud. The company will then obliviously deliver the identity threshold representation sets for the thresholds of the decisional branching nodes and the indexes of the concerned attributes to TA so that TA can generate the ReKeys corresponding to the rest clients in the system using the key private re-encryption scheme. Since the final scheme is based on the newly proposed key private proxy re-encryption scheme, we will present this scheme first.

ReKey(id1, id2, msk): This algorithm is performed by TA. Upon receiving the request from delegator *D* of re-encryption from id1 to id2, it first runs the Ext algorithm on id2 to generate skid2 . Then it outputs the re-encryption key from id1 to id2:

$$\begin{aligned} rk_{\mathsf{id}_1,\mathsf{id}_2} &= (rk^{(1)}_{\mathsf{id}_1,\mathsf{id}_2}, rk^{(2)}_{\mathsf{id}_1,\mathsf{id}_2}) \\ &= (H_1(\mathsf{id}_1)^s \cdot g^{H_2(\mathsf{sk}_{\mathsf{id}_2}||N_{\mathsf{id}_1,\mathsf{id}_2})}, N_{\mathsf{id}_1,\mathsf{id}_2}) \end{aligned}$$

where *N*id1,id2 is a random element from G. If *C* is a re-encrypted ciphertext (c' 1, c2, c' 3, c4) (assume that the receiver of the re-encrypted ciphertext is id ′),Compute

$$\begin{aligned} & H_4\left(\frac{c'_3}{c'^{H_2(sk_{\mathsf{id}'}||c_4)}_1}\right) \oplus c_2 \\ = \; & H_4\left(\frac{e(y,H_1(\mathsf{id})^r)\cdot e(g,g)^{r\cdot H_2(sk_{\mathsf{id}'}||N_{\mathsf{id},\mathsf{id}'})}}{(e(g,g)^r)^{H_2(sk_{\mathsf{id}'}||N_{\mathsf{id},\mathsf{id}'})}}\right) \\ & \oplus(\sigma||m) \oplus H_4(e(H_1(\mathsf{id}),y)^r) = \sigma||m \end{aligned}$$

The last step can be omitted if only chosen ciphertext attack (CPA) security is considered. The CPA security 2 is sufficient in practice assuming there is a secure and authenticated channel between the company and the cloud.

of the ciphertexts in the encryption schemes guarantee that the cloud can neither find out the information attached to the leaf nodes nor the order or the thresholds of intermediate branching nodes. The key privacy guarantees that the cloud obtains no useful information on the branching program while completing all the computationally intensive encryption operations for the company. As in the first improvement, the transformation key contains no information on a client's query vector v due to the mask privacy, which defeats the cloud's attack through the identity testing.

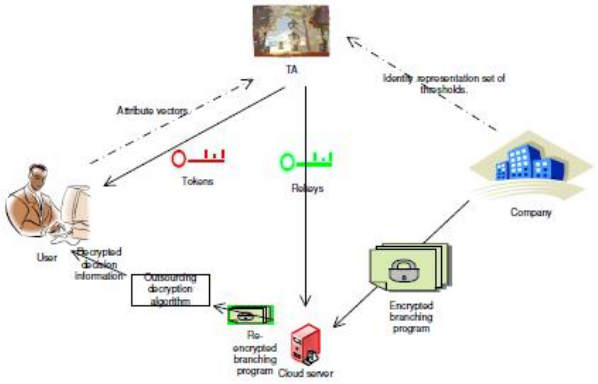

Fig. 5. Final CAM with Full Privacy and High Efficiency

The client delivers his index *i* to the cloud which will then return the respective ciphertext. The client can either download all the ciphertexts and transformation keynband perform the rest decryption steps, or he could start tomnrun Dec(skid,*C*id), where id *S*[0;*t*1+_*i*1] or *S*[*t*1+_*i*1+1;*Max'*] to decrypt from p1 and then download the ciphertext and themtransformation key for the next node according to the decryption result. If he chooses the latter approach, then he only needs to access the ciphertext corresponding to a path from the root node to a leaf node instead of all the ciphertexts for all nodes in the directed branching tree. However, in so doing, the client has to access the cloud multiple times proportional to the length of the path. Compared with the first improvement, the cloud does not need to perform any computation when it interacts with the client in this case because the client alone can complete all the necessary decryption steps. On the other hand, the client does not need to compute any bilinear map since the bilinear operation has already been completed by the cloud due to the preprocessing step in the ReEnc(*C*id1 , *rk*id1;id2) algorithm as shown in subsection IV-C1.

IV.SECURITY ANALYSIS AND PERFORMANCE EVALUATION

In our CAM, we observe that the cloud obtains no information on either the individual query vector v or the company diagnostic branching program as in our first improvement.The cloud obtains no information on the company's branching program due to the semantic security of the proxy reencryption .

Efficiency.

To assess our CAM, we conduct a few experiments. We used a laptop with a 2.4 GHz processor with a 4GB of RAM to

simulate the cloud server and the company, and 1 GHz AMR-based iPhone with 512MB RAM to simulate a client. All the timing reported below are averaged over 100 randomized runs. We assume a maximum of $k = 1000$ nodes in the branching program, which can express most complicated decision support systems as used in the MediNet with 31 nodes (Fig. 2). The attribute vector has a maximum of $n = 50$ attributes, which contain much richer information than the MediNet project with four attributes. We use the benchmark results from the PBC library for our evaluation. Most of current private telemonitoring schemes are based on anonymization, which are ineffective as we alluded before. Another line of work focuses on privacy preserving diagnostic programs. At the end of protocol run, a client obtains nothing on the diagnostic program but the diagnostic result while the company obtains no information on the client's private data. Since our application scenario assumes the clients hold relatively resource-constrained mobile devices in a cloud assisted environment, it would be helpful if a client could shift the computational workload to the cloud. However, there seems no trivial approach to outsourcing the decryption of garbled circuit currently. Our proposed system adopts the recently proposed decryption outsourcing to significantly reduce the workload of both the company and clients by outsourcing the majority of the computational tasks to the cloud while keeping the company offline after the initialization phase.

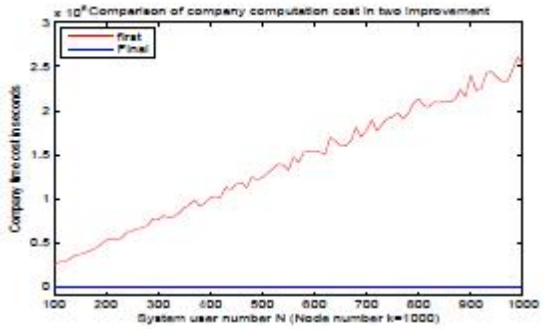

Fig. Comparison of company computations in our two improved CAM designs

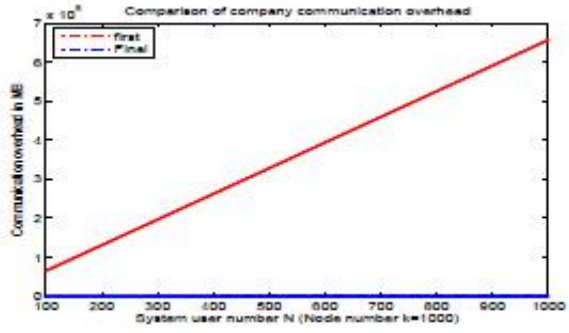

Fig. Comparison of company communication overheads in our two improved CAM designs

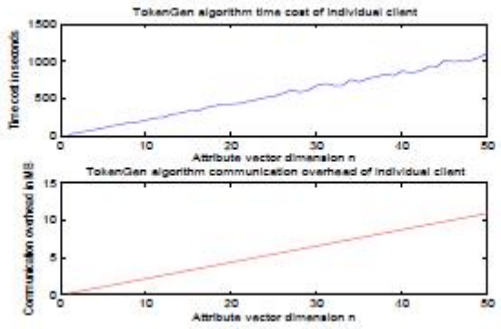

Fig. Workload of Individual Token Generation

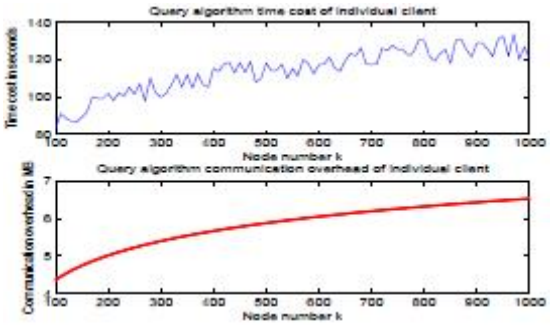

Fig. Workload of Individual Query

CONCLUSION

we design a cloud-assisted privacy preserving mobile health monitoring system, called CAM, which can effectively protect the privacy of clients and the intellectual proerty of mHealth service providers. To protect the clients' privacy, we apply the anonymous Boneh-Franklin identitybased encryption (IBE) in medical diagnostic branching programs. To reduce the decryption complexity due to the use of IBE, we apply recently proposed decryption outsourcing with privacy protection to shift clients' pairing computation to the cloud server. To protect mHeath service providers' programs, we expand the branching program tree by using the random permutation and randomize the decision thresholds used at the decision branching nodes. Finally, to enable resource constrained small companies to participate in mHealth business,our CAM design helps them to shift the computational burden to the cloud by applying newly developed key private proxy re-encryption technique. Our CAM has been shown to achieve the design objective.

REFERENCES

[1] P. Mohan, D. Marin, S. Sultan, and A. Deen, "Medinet: personalizing the self-care process for patients with diabetes and cardiovascularndisease using mobile telephony." *Conference Proceedings of the* n*International Conference of IEEE Engineering in Medicine and* n*Biology Society*, vol. 2008, no. 3, pp. 755–758. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19162765

[2] A. Tsanas, M. Little, P. McSharry, and L. Ramig, "Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests,"n *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 4, pp. 884–893, 2010.

[3] G. Clifford and D. Clifton, "Wireless technology in disease management and medicine," *Annual Review of Medicine*, vol. 63, pp. 479–492, 2012.

[4] L. Ponemon Institute, "Americans' opinions on healthcare privacy, available: http://tinyurl.com/4atsdlj," 2010.

[5] A. V. Dhukaram, C. Baber, L. Elloumi, B.-J. van Beijnum, and P. D. Stefanis, "End-user perception towards pervasive cardiac healthcare services: Benefits, acceptance, adoption, risks, security, privacy and trust," in *PervasiveHealth*, 2011, pp. 478–484.
[6] M. Delgado, "The evolution of health care it: Are current u.s. privacy policies ready for the clouds?" in *SERVICES*, 2011, pp. 371–378.
[7] N. Singer, "When 2+ 2 equals a privacy question," *New York Times*, 2009.
[8] E. B. Fernandez, "Security in data intensive computing systems," in *Handbook of Data Intensive Computing*, 2011, pp. 447–466.
[9] A. Narayanan and V. Shmatikov, "Myths and fallacies of personally identifiable information," *Communications of the ACM*, vol. 53, no. 6, pp. 24–26, 2010.
[10] P. Baldi, R. Baronio, E. D. Cristofaro, P. Gasti, and G. Tsudik, "Countering gattaca: efficient and secure testing of fully-sequenced human genomes," in *ACM Conference on Computer and Communications*n *Security*, 2011, pp. 691–702.
[11] A. Cavoukian, A. Fisher, S. Killen, and D. Hoffman, "Remote home health care technologies: how to ensure privacy? build it in: Privacy by design," *Identity in the Information Society*, vol. 3, no. 2, pp. 363–378, 2010.
[12] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 111–125.
[13] ——, "De-anonymizing social networks," in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2009, pp. 173–187.
[14] I. Neamatullah, M. Douglass, L. Lehman, A. Reisner, M. Villarroel, W. Long, P. Szolovits, G. Moody, R. Mark, and G. Clifford, "Automated de-identification of free-text medical records," *BMC medical informatics and decision making*, vol. 8, no. 1, p. 32, 2008.
[15] S. Al-Fedaghi and A. Al-Azmi, "Experimentation with personal identifiable information," *Intelligent Information Management*, vol. 4, no. 4, pp. 123–133, 2012.
[16] J. Domingo-Ferrer, "A three-dimensional conceptual framework for database privacy," *Secure Data Management*, pp. 193–202, 2007.
[17] T. Lim, *Nanosensors: Theory and Applications in Industry, Healthcare,and Defense*. CRC Press, 2011.
[18] X. Zhou, B. Peng, Y. Li, Y. Chen, H. Tang, and X. Wang, "To release or not to release: evaluating information leaks in aggregate human-genome data," *Computer Security–ESORICS 2011*, pp. 607–627, 2011.
[19] R. Wang, Y. Li, X. Wang, H. Tang, and X. Zhou, "Learning your identity and disease from research papers: information leaks in genome wide association study," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 534–544.
[20] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization," *UCLA Law Review*, vol. 57, p. 1701, 2010.
[21] P. Institute, "Data loss risks during downsizing," 2009.
[22] P. Dixon, "Medical identity theft: The information crime that can kill you," in *The World Privacy Forum*, 2006, pp. 13–22.
[23] K. E. Emam and M. King, "The data breach analyzer," 2009, [Available at: http://www.ehealthinformation.ca/dataloss].

Mr. S.Manjunath is a student of Narayana Engineering College,Gudur, presently he is pursuing M.Tech (C.S) and his areas of interest are Cloud Computing and Secure Computing.

Mr. N.Koteswara Rao, well known author and excellent teacher received M.Tech (C.S.E) and he is working as H.O.D for the department of Computer Science & Engineering in Narayana Engineering College Gudur. He has vast teaching experience in various engineering colleges. He has couple of publications both National & International Conferences / Journals. His interested research on Cloud Computing ,Data warehousing and data mining, information security,Data communications and networks.