Information Systems 48 (2015) 132–150

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Collaborative and secure sharing of healthcare data in multi-clouds

Benjamin Fabian [a,*], Tatiana Ermakova [b], Philipp Junghanns [a]

[a] Humboldt-Universität zu Berlin, Institute of Informations Systems, Spandauer Str. 1, 10178 Berlin, Germany
[b] Technical University of Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany

ARTICLE INFO

Available online 29 May 2014

Keywords:
Cloud computing
Healthcare
Security
Privacy

ABSTRACT

In healthcare, inter-organizational sharing and collaborative use of big data become increasingly important. The cloud-computing paradigm is expected to provide an environment perfectly matching the needs of collaborating healthcare workers. However, there are still many security and privacy challenges impeding the wide adoption of cloud computing in this domain. In this paper, we present a novel architecture and its implementation for inter-organizational data sharing, which provides a high level of security and privacy for patient data in semi-trusted cloud computing environments. This architecture features attribute-based encryption for selective access authorization and cryptographic secret sharing in order to disperse data across multiple clouds, reducing the adversarial capabilities of curious cloud providers. An implementation and evaluation by several experiments demonstrate the practical feasibility and good performance of our approach.

1. Introduction

The so-called "big data era" [1] is characterized by large amounts of data being collected and stored for years, as well as by new techniques that are enabling organizations to handle and analyze them. These developments are also increasingly important for medicine and healthcare delivery. Medical records in electronic form provide multiple benefits such as flexible handling, searching, as well as better decision making and analytics when being shared and collaboratively used among healthcare-related parties [2–6]. But even for a single patient only, such records may reserve large storage capacities of a health center [7], and currently emerging applications such as sensor-supported telemedicine will increase this trend.

Cloud computing technology [8] perfectly matches such "big data" challenges by providing nearly unlimited storage resources on demand [1]. In healthcare, it is also gaining particular popularity by facilitating an inter-organizational medical data sharing environment [9–12]. On the other hand, this paradigm also involves many security and privacy risks that lead to concerns among patients and medical workers [13–15] who are being particularly afraid of losing the control over sensitive medical records while storing them on not fully trusted third-party servers [16–18]. Regulations such as HIPAA also call for a strong protection of medical records [19].

In this paper, we present an architecture for secure and privacy-preserving sharing of medical big data between different cooperating organizations in a data-cloud environment, which consists of multiple, independent cloud providers ("Multi-Cloud"). Data-cloud providers are assumed to be semi-trustworthy: honest in securing the services they provide against external adversaries, but curious with respect to the data that they store. In our architecture, medical records are created, maintained and retrieved by

* Corresponding author.
E-mail addresses: bfabian@wiwi.hu-berlin.de (B. Fabian), tatiana.ermakova@tu-berlin.de (T. Ermakova), philipp.junghanns@gmail.com (P. Junghanns).

http://dx.doi.org/10.1016/j.is.2014.05.004
0306-4379/

authorized users in cooperating health centers. Mediating Multi-Cloud Proxies will distribute and retrieve encrypted medical records to and from multiple data clouds in parallel. In order to further protect the data from curious cloud providers, we adopt a secret-sharing approach proposed by Krawczyk [20], apply it to electronic medical records, and distribute the resulting shares to different independent cloud providers. Moreover, this approach can provide increased availability of medical records by providing a larger set of fragments to reconstruct the documents from. For unlinkability of shares to patient identifiers, we provide a method for constructing external identifiers by a cryptographic hash function. In order to provide selective sharing of data among different groups of users, our architecture supports role-based access policies for selected attributes or sections of a medical record, enforced by attribute-based encryption [21]. Our current paper extends an earlier high-level sketch of our architecture [22] by a much more sophisticated and refined design, a new and complex implementation, and several performance experiments.

The current work is part of an ongoing larger engineering project in healthcare, the so-called TRESOR (TRusted Ecosystem for Standardized and Open cloud-based Resources) research project [30]. It is conducted in accordance with design science frameworks such as those proposed by [23,24]: We identify and motivate the problem in Section 2, give an overview on related work in Section 3, and define the objectives of the solution in Section 4. The main architecture design is presented in Section 5. Details on our implementation and experimental evaluations are given in Section 6.

2. Collaborative big data in healthcare

The big data concept in itself is not new in medicine and healthcare delivery. Healthcare providers deal with large amounts of medical records, which are not just improving in their quality and detail, but also are continuously increasing in size due to technological advances. Such records have to be archived in the long term even after accomplishing the patient's treatment. Nowadays, these may imply a several MB image and a several hundred MB video per single patient [7]. Sensors can also be adopted for enabling different healthcare-related scenarios, e.g., to automatically gather patient's data [3,25], to monitor hemoglobin concentration changes in the brain and tissues [26], to support elderlies in their self-activities and patients needing a physiological control [27], as well as to support patients suffering from depression [28].

Medical big data generates special value when being shared and collaboratively used among different parties involved in the healthcare area (e.g., healthcare centers, laboratories, pharmacies, patients, health insurance, quality assurance in healthcare service delivery, researchers, national and regional health authorities). Many researchers [2–6,18] as well as stakeholders we interviewed consider immediate access to previously generated medical records during healthcare service delivery as highly important.

The diagram in Fig. 1 (in Business Process Modeling Notation, BPMN [29]) provides an example scenario of inter-organizational data sharing. After being treated in the health center, a patient is transferred to a specialized

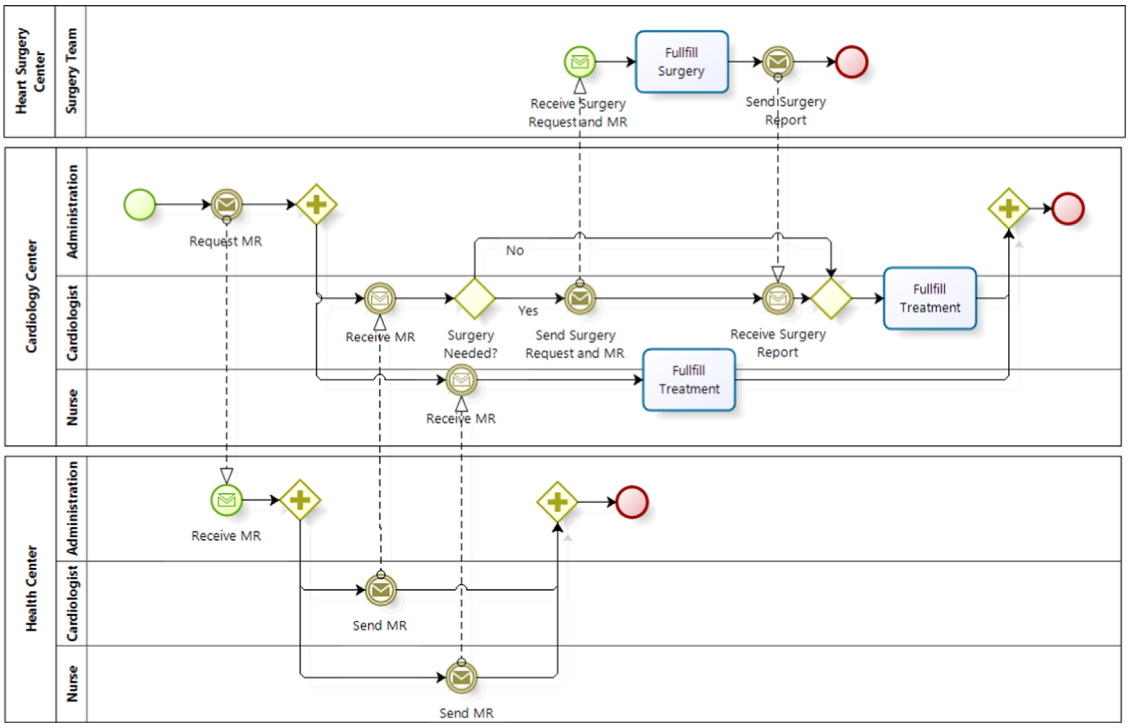

Fig. 1. An example scenario of inter-organizational medical record sharing.

cardiology center. Before starting the treatment, the cardiology center acquires the previous medical records, e.g., laboratory tests, images, videos, diagnoses and medication, which have been generated by the initial health center. The data being transmitted among institutions are usually part of bilateral agreements and have to correspond to various access policies in accordance with the currently existing medical regulations such as HIPAA. Should a surgery be performed, the cardiologist of the cardiology center places a surgery request with the surgery center providing a specially generated medical record and gets a surgery report thereafter.

Inter-organizational data sharing is supported by the notions of EHR (electronic health record), EMR (electronic medical record) and PHR (personal health record). Exemplary application scenarios are presented by the TRESOR research project [30], as well as by proposals of centralized emergency medical systems [2,5,6] and Picture Archiving and Communication Systems (PACS) [31,4,32]. Another recent study [33] identified several projects to reshape medicine and healthcare delivery through collaborative analytics of medical big data. The application fields ranged from disease detection, disease outbreak prediction, and choice of a therapy, to useful information extraction from clinical notes, and medical data gathering and organizing. The ideas of collaborative big medical data analytics are also followed by research projects such as Cloud4health, aimed at the evaluation of clinical data for clinical trials and registers [34], and GeneCloud directed at development of medical therapies [35].

In medicine and healthcare delivery, authors often refer to the preferable storage capacities offered by cloud computing technology [13,9]. This is demonstrated in the approach by [36], turning small hand-held mobile devices into “powerful workstations” by delegating management of archives, preprocessing and rendering of medical images to the cloud. The Cloud4health and GeneCloud research groups also rely on cloud computing when adopting text analytics, data warehousing, and simulations. Further examples may be found in [3,26,25]. In healthcare, cloud computing is also becoming more important as facilitating inter-organizational medical data sharing such as Microsoft HealthVault, Google Health, WebMD, Georgia Tech's MedVault or Harvard's Indivo, see also [9–12].

3. Related work

Security and privacy for cloud computing in general is a large research area. In this paper, we can only refer to some important related works: the guidelines on cloud computing security by ENISA [37] and NIST [38], as well as a recent article summarizing the state of the art [39], where also a set of interesting security measures such as a cloud auditing service are presented that may be combined with our approach. In contrast to this article, we focus on leveraging multi-cloud settings with secret-sharing and use different cryptographic measures.

For cloud computing in healthcare, there are also many approaches aimed at mitigating security and privacy risks, which have been reviewed in our previous research [40,22]. For the usage of cloud computing in healthcare, [41] presents a secure e-health infrastructure based on Trusted Virtual Domains. [42] establishes trustworthy middleware services with the goals of security, privacy, and resilience. [43] elaborates on unlinkability between patients and medical records. [11,16,15,44,17,14,45] leverage cryptographic access-control schemes to electronic health records, while [43] enables a keyword search over encrypted documents.

In contrast to previous approaches, our system is also not just based on a single encryption algorithm such as in [17,14,45,18,46], where cryptographic implementations may contain bugs or secret keys could be compromised at any time in the future, which would allow curious cloud providers immediate access to the data. In our approach, secret sharing provides an additional protection in those cases. The approach introduced by [47] shows some similarities to our architecture, but focuses on availability and therefore lacks any of the security measures that our solution provides, such as cryptographic and role-based access-control mechanisms.

4. Security and privacy requirements

Besides its benefits, the new promising cloud computing paradigm implies many security and privacy risks that could impede its wide acceptance among patients and medical workers [13–15]. The particular concern is about losing control over sensitive medical records while storing them on a third-party server outside of a trust boundary [16–18]. Our architecture aims to address many of the following related security and privacy requirements, which have been collected by an earlier literature survey and also from stakeholder interviews [22,40]. We acknowledge that there will be many interactions between those requirements. A primary concern for all stakeholders involved is the secure and privacy-preserving storage and handling of medical records (MRs). This involves the following major aspects.

Confidentiality of medical record content: The highly sensitive content of medical records needs to be protected against any external entities, such as Internet or cloud providers, but also against unauthorized internal personnel. This requirement can be refined to particular attribute values in case the MR contains relational data.

Confidentiality of medical record existence: Even the knowledge of the existence of a medical record could constitute a privacy risk if such knowledge combined with adversarial background knowledge could enable certain inferences. In this paper, we explicitly reduce the set of potential adversaries for this particular requirement to external parties, since corresponding countermeasures against internal health personnel could be too restricting in practice.

Anonymity of patient in a medical record: Identifiers of patients, such as names and social security numbers, need to be treated as confidential wherever possible. This is especially important against external parties, but if possible also internal employees when exact patient identifiers are not needed. In this paper, we cannot discuss the problem of so-called *quasi-identifiers*, i.e., sets of apparently “harmless”

attributes that in combination could identify an individual. Integrating such aspects in future work will involve research on data anonymization and anonymous data publishing, starting from early concepts such as k-anonymity up to more recent proposals such as differential privacy.

Unlinkability between medical records and patients: Even if patient identifiers and medical records are leaked to unauthorized internal or external parties, they should not be able to link a medical record with a particular patient. This requirement has a high interdependency with the earlier ones.

Integrity and authenticity of medical records: Medical records should not be modified by unauthorized parties or random errors during transfer and storage in the clouds, and if they are, such modifications need to be detectable. In addition, the origin and authorship of MRs need to be verifiable.

Availability of medical records: Whenever requested, the MR should be retrievable by authorized entities, without larger delays. This requirement could include the need for access to MRs without depending on direct patient involvement, a situation we assume in this paper. This does not necessarily mean that a patient would not be involved in the formulation of access policies, only that in a concrete enforcement situation his or her direct and online interaction is not needed. In complex system landscapes, "enough" measures should be established to increase MR availability to an acceptable level, and to reduce bottlenecks and single points of failure where possible.

We will now turn to more architecture-related requirements concerning security and privacy. This involves the users of the architecture (such as medical personnel), but also serves to provide security to the general environment in order to improve MR security as well.

Authentication of users and services: Users who would like to retrieve or store medical records in the clouds need to be authenticated. All distributed system components of the architecture, such as web services of proxies or cloud providers should also provide authentication.

Fine-grained authorization and access control: Based on their authenticated identities, users should be provided with authorization to access services of the architecture, as well as for accessing and modifying health records, or fine-grained, even particular attributes.

Anonymity of user identities: Following the paradigm of multilateral security [48], which is respecting the requirements of multiple stakeholder of a system, system users such as healthcare personnel should remain anonymous, at least against external parties. Internally, this requirement may get in conflict with the following internal audit requirements.

Auditing capability: Actions of users with respect to health records should be monitored and logged by authorized system components in order to provide a way to track internal violations of patient privacy or system security.

Confidentiality of user access privileges: Against external parties, user access privileges should be kept confidential since on the one hand this could affect a user's privacy, and on the other may also leak valuable information to an adversary.

Access right revocation: If necessary, authorization such as access rights of users to MRs should be revocable.

Emergency exceptions: If the deployment scenario also involves emergency data access, e.g., for mobile health workers who may have no established or individual authorization to a particular patient's MR, an emergency access to that data should be implemented if this is agreed to by all stakeholders.

Scalability of security measures: The architecture and in particular all security measures should scale well in at least two dimensions: (i) amount of data such as size and number of records, up to big data, and (ii) number of participating health centers and users.

Availability of systems: Components of the architecture should be available whenever needed by an authorized user.

Archiving of medical records: The architecture should provide means for long-term availability and preservation of MRs if this is in scope with patients' privacy preferences and legal requirements.

Efficiency and usability: All security components should be as efficient and usable as possible, not least in order to reduce the risk of being circumvented.

5. Architecture

5.1. Architecture overview

In the following, we will give an overview on our architecture. A high-level view of the *Basic Architecture* is shown in the large box of Fig. 2. At the top of the figure we see a patient visiting three different health centers (HCs) over time. During each visit, new data about the patient is generated and stored in separate medical records, which are identified by a common identification scheme.

At HC A, the client software of a doctor signs and encrypts the MR (Step 1) and sends it to the local Multi-Cloud Proxy (MCP) in Step 2. The proxy splits the encrypted MR according to a secret-sharing scheme, and distributes each share over the Internet to a different Cloud Provider (CP) in Step 3. External identifiers for shares are constructed in such a way that later, authorized clients can calculate the same identifiers and retrieve the data using an analogous procedure. Later, the patient visits HC B, which could involve a medical specialist for a different health problem. Here, a new MR specific to the visit is created, encrypted (Step 4), and sent to the local proxy of HC B (Step 5). Analogously to Step 3, the proxy splits the encrypted MR and distributes the shares to different Cloud Providers (Step 6) using a common identifier scheme.

The retrieval process is similar to the storage process. At a later time, the patient visits a third health center, C. Here, a doctor is interested in the full medical history of the patient and requests medical records of HC A and B from the local Multi-Cloud Proxy at HC C. The proxy retrieves enough corresponding shares from the cloud providers (Step 7). This retrieval involves authentication to the cloud providers. Then, the proxy reconstructs the encrypted MRs (Step 8). Both records are sent back to the doctor's client software, which decrypts them and verifies their authenticity (Step 9).

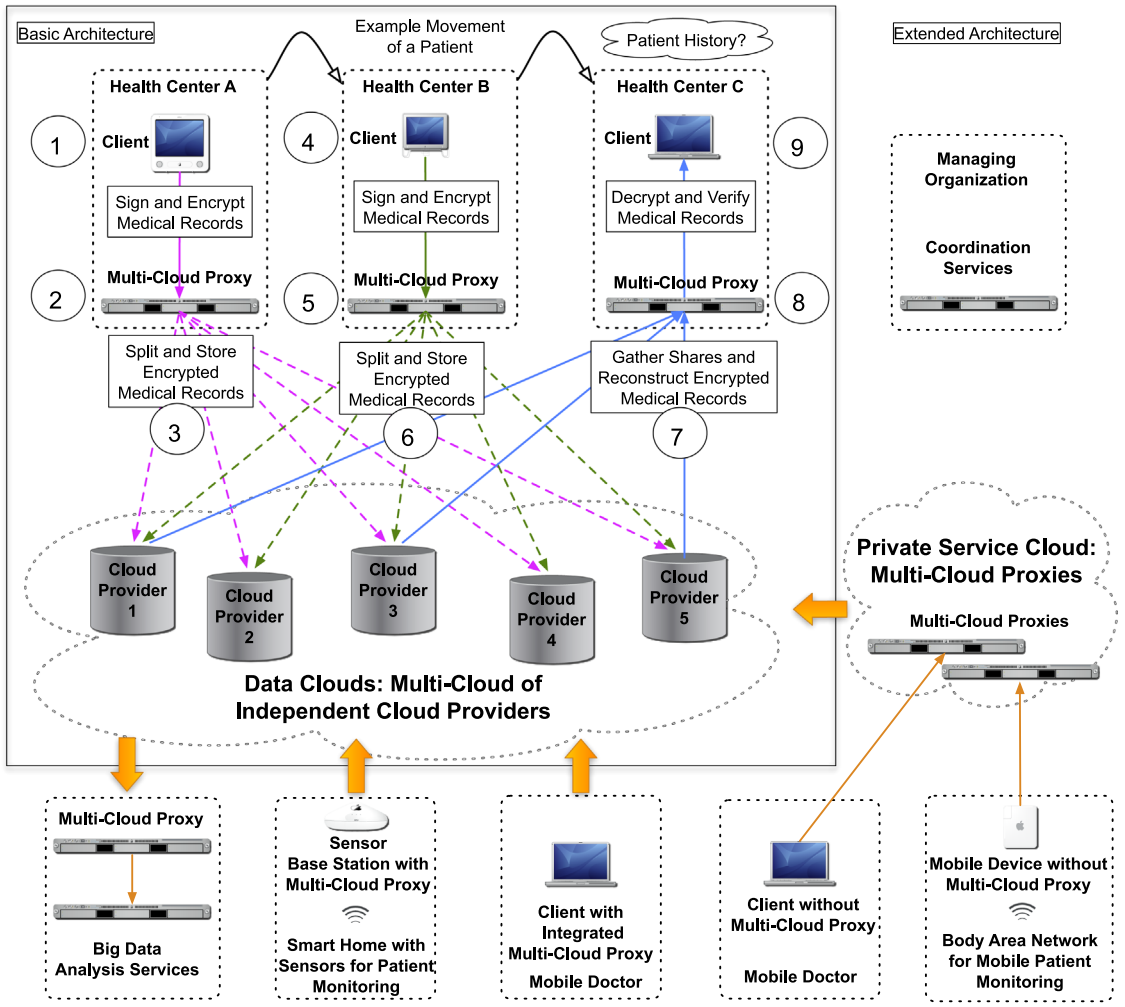

Fig. 2. Architecture overview.

An extended version of our architecture (Fig. 2, *Extended Architecture*) could also enable monitoring of patients by sensor-equipped smart homes, or mobile patient monitoring using body-area sensor networks or smart phones. This extended architecture also features mobile health workers who could retrieve data from the clouds in order to conduct diagnoses, and authorized clients that are conducting advanced analytics on the huge amount of medical data that is available in the multi-cloud. In order to support clients with low-performance hardware not supporting the cryptographic operations of the MCP, there could be a private Service Cloud where this proxy functionality is offered as a service to authorized clients. This Service Cloud needs strong security measures to reduce the risk of compromise, but could then also be used for redundancy of MCPs in the basic architecture.

We note that the extended architecture will bring about challenges to scalability in both dimensions: the number of devices and the amount of data involved. Moreover, the scalability of managing access-control policies and cryptographic keys are important issues. In the current prototype and discussion presented later in the paper, we will mainly focus on the basic architecture shown in the box of Fig. 2, but will point out important directions for future work in order to realize the extended scenario.

5.2. Security and privacy measures

In our approach, we assume that all participating organizations, such as HCs or CPs, have a common interest in securing the infrastructure and data against external, third party adversaries. Hence, the establishment of common and cooperative security mechanisms will be feasible, even though many practical and procedural challenges could arise when implementing them in concrete usage scenarios. We acknowledge these important challenges, but consider them out of scope of our current paper. Currently, we also assume that data owners are the health centers or particular health workers. We leave an extension of this assumption to multilateral security requirements and in particular the challenges of patient-centric data management for future work. We present an overview of security and privacy measures in Table 1.

Table 1
Main security and privacy measures.

Number	Measure for security and privacy	Goal	Location
1	Client and service authentication	Prevent unauthorized participation in the system	All participants
2	Network security, TLS	Prevent communication eavesdropping or modification	All participants
3	Federated identity management	Increase usability and reduce management overhead	HCs
4	Access control to cloud data	Coarse-grained: prevent unauthorized retrieval of MRs from the clouds	CPs
5	Access control to MR (RBAC)	Fine-grained: protect access to sensitive information in MRs	HCs
6	Attribute-based encryption (ABE)	Protect access to sensitive information in MRs	Authors, HCs
7	Digital signatures, HMAC	Prevent unauthorized modification of MRs and shares	Authors, HCs
8	Secret-Sharing of MRs	Increase availability of data. Prevent spying on data	HCs
9	Cryptographic hash function for external identifiers	Anonymity, unlinkability between MRs and patient IDs	HCs
10	Data Replication at CP	Prevent data loss	CPs

Since we assume a collaborative healthcare scenario, a cooperative infrastructure for Client and Service Authentication should be practically feasible. This could involve a central Certificate Authority (CA), a tree or forest of CAs forming a Public-Key Infrastructure (PKI) or a fully connected Web-of-Trust between participating organizations [49]. Any client program or service should be authenticated, preventing unauthorized third parties from taking part in the system simply by adopting a false identity.

Second, it is necessary that classical network security protocols be in place, which prevent eavesdropping or forging of any communication by third-party adversaries. Depending on the concrete realization of communication between partners, such protocols could include Virtual Private Networks (VPNs) between all cooperating partners, Transport Layer Security (TLS) [50] including HTTPS for Web-based information exchange [49] or, more advanced, Web Service Security protocols, if a cooperative Service-Oriented Architecture (SOA) is used [51].

As a recommended building block, Federated Identity Management and User Authentication could increase the usability of the system by providing a common view on user identities across organizational borders [52]. Health centers cooperate by implementing local user identification and sharing authentication status-information according to a mutual trust relationship. Moreover, authenticated users at HCs can be authorized to access documents stored at the CPs, which can be implemented by including the CPs as consumers of the federated identity management and authentication process.

Authentication and (possibly federated) identities would serve as prerequisite for authorization. Here we propose that a framework of access-control policies authorizes participating HCs and their employees, and possibly also authorized external information clients. In this paper, we assume Role-Based Access Control (RBAC), since we expect clear correspondences between job roles and information demands in a HC [53].

Depending on the granularity of access control, such policies can be technically enforced by access control mechanisms at HCs and CPs, but also by implementing advanced encryption methods that are operating on the documents stored in the clouds. We adopt a combination of both approaches: first, accessing any document in a participating cloud should be possible only for authorized clients. If additional advanced encryption methods are applied, this first line of access control could be coarse-grained, reducing the overhead (and possible information leakage) of communicating fine-grained policies to the CPs. For example, in order to retrieve an encrypted document, proxies may only need to provide proof that they are part of an authorized HC participating in the collaboration. Fine-grained access control by encryption could enforce that only truly authorized individuals could decrypt sensitive information included in this document. Recent advances in this direction include Attribute-Based Document Encryption (ABE), which allows fine-grained access control by encryption at the level of data attributes [17,14].

In order to provide protection against unauthorized modification of medical records, we use digital signatures issued by authors of each organization. These signatures are applied at least at the document level before documents are stored in the clouds, but can also be added at the level of sections or single attributes of the MR. Not least, every CP should provide internal redundancy and backup mechanisms against loss of documents, in order to achieve long-term availability of information. In addition, each HC needs to adopt procedures for storing and conserving cryptographic parameters and keys.

As a further contribution, we utilize secret-sharing schemes in order to reduce the risk of information leakage in multi-clouds even further, in particular to mitigate potential encryption software errors or compromised decryption keys, in the face of curious or hacked cloud providers. Moreover, such schemes could also increase the availability of data stored in the clouds. As a side note motivated by national laws and politics, share distribution and threshold selection may also be refined to control the number of shares that get distributed to cloud providers outside of a certain area of legislation (such as EU or U.S.), and prevent foreign providers from reconstructing the MR even if they all collude or are forced to pool their shares by decree of a government.

In order to provide a fast verification mechanism for share integrity and authenticity, two different cryptographic Hash-based Message Authentication Codes (HMACs) [54]

are applied that differ in the secret key used. One HMAC can be verified by all proxies after collecting the shares, but is less secure against forgery. The second HMAC allows for strong authenticity protection, but if no further sharing of the key is organized, it can only be verified by the storing Multi-Cloud Proxy. We emphasize that the main and final protection of MR integrity and authenticity are the digital signatures at document or attribute-level. In order to identify, store, and retrieve shares, we also present a way to construct confidentiality-preserving external share identifiers from potentially sensitive internal identification schemes.

In summary, a plethora of security measures will be deployed in our architecture. In practice, there should also be a set of Coordination Services that support the cooperative efforts to utilize the multi-cloud and to provide inter-organizational measures for security and privacy. Besides credential management and performance monitoring of the multiple clouds, these services should include a Certificate Authority and some supporting services for federated identity management and collaborative RBAC policies. Moreover, public parameters of cryptographic schemes need to be stored, managed, and communicated to the cooperation partners, e.g., if it is required to flexibly change and communicate parameters at run-time. Moreover, the use of ABE involves a high amount of cooperation and a trusted common master key, if this approach should scale to an inter-organizational setting. As a working assumption, we expect those services to be centralized, though not necessarily be controlled by a single stakeholder, in order to prevent too strong concentrations of trust and power. To what extent each single service could also be designed in a decentralized or even fully distributed fashion is an important challenge for future research.

The storage process from a single HC, including security measures, is shown in Fig. 3, where all internal and external communication is assumed to be additionally secured by TLS. Details of selected measures will be discussed in the next sections.

5.2.1. RBAC

In a Role-Based Access Control (RBAC) model, privileges are not directly granted to users, but to roles, e.g., abstract and stable job positions of users having a similar function in the organization [55,53]. This reduces the management overhead of formulating and managing access control policies compared to a per-user access control list, and enables scalability of such policies for large user bases. In particular if combined with role hierarchies, RBAC is well suited for inter-organizational settings, though coordinated processes to create and manage collaborative policies are still an active topic of research [56]. An example excerpt of an RBAC policy, which corresponds to some roles in the use case of the BPMN diagram of Fig. 1, could be expressed in an informal notation as follows:

Role := (Cardiologist in HealthCenter)
Allowed Action := (Read OR Write Access)
Target := (MR Patient 2342, Section: LaboratoryResults)

Role := (Cardiologist in HeartCenter)
Allowed Action := (Read Access)
Target := (MR Patient 2342, Section: LaboratoryResults)

5.2.2. Attribute-based encryption

In our architecture, we enforce the role-based access control mechanism cryptographically based on the ciphertext policy attribute-based encryption (CP-ABE) recently developed by Bethencourt, Sahai and Waters [21]. CP-ABE was also applied for secure health record storage by [57,18,46]. In CP-ABE, the encrypting party combines the encrypted data with an access control policy, which ranges over user attributes and defines who can decrypt them. Instead of authorizing users, we grant permissions to roles in order to provide a bridge from RBAC policies, and therefore we use attributes describing roles.

For instance, the cardiologist of the health center transmitting the MR to the cardiologist of the heart center in Fig. 1 may specify an RBAC policy that gets translated to an ABE access policy for the MR as follows: *HeartCenter AND Dept: AdultCardiology AND Doctor AND Cardiologist*. If she additionally provides access to the surgeon in the Heart Surgery Center, the access policy may look like: (*HeartCenter AND Dept: AdultCardiology AND Doctor AND Cardiologist*) *OR* (*HeartSurgeryCenter AND Dept: AdultCardiology AND Doctor AND Surgeon*). Providing tool-support for such mappings, as well as to investigate practical limitations, are goals of future work.

5.2.3. Secret sharing

In this paper, we focus on Shamir's secret-sharing scheme [58] and Rabin's Information Dispersal Algorithm

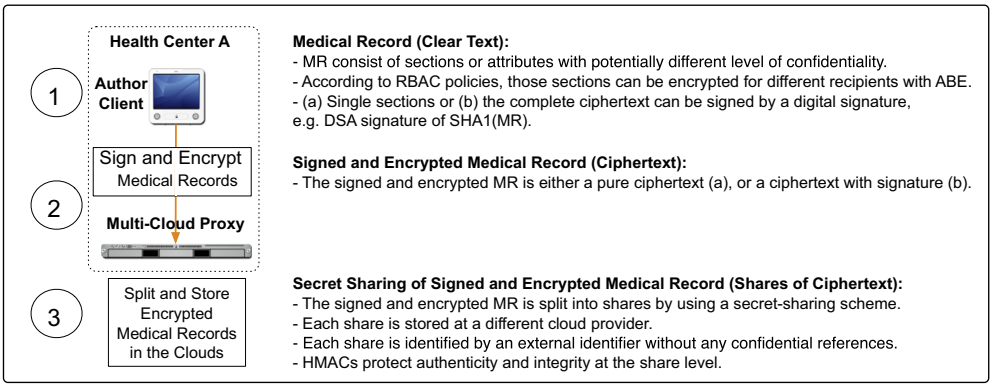

Fig. 3. Details of storage process.

(IDA) [59] because both schemes are time-tested. Both schemes represent a so-called (t, n)−threshold scheme with $1 \le t \le n$, which is a protocol for the distribution of a secret document D among n parties such that recovery of the document is possible in the presence of at least t shares for a fixed value t, $1 \le t \le n$, while fewer shares give no information about the secret document, providing *perfect secrecy* as Shamir's secret-sharing scheme, or no computationally extractable information, providing *computational secrecy* as Rabin's information dispersal algorithm.

Suppose, data D is (or can be) represented as a number. Then Shamir's secret-sharing algorithm sets $a_0 = D$, chooses elements $a_1, \ldots, a_{t-1}$ at random, takes any distinct values $x_1, x_2, \ldots, x_n$ with $n > t-1$ and computes the shares as $s_i = (x_i, f(x_i) = a_0 + a_1 x_i^1 + \cdots + a_{t-1} x_i^{t-1})$ for $i = 1, \ldots, n$. For reconstructing the document, only the non-zero coefficient, i.e., a_0, has to be computed, which is the value of $f(x)$ in $x=0$. The *Lagrange* form of the interpolating polynomial is typically used, given by:

$$L(x) = \sum_{i=1}^{t} y_i \prod_{j=1, j \ne i}^{t} \frac{x_j - x}{x_j - x_i}.$$

Rabin's IDA on the other hand splits D into blocks of length t, i.e., $D = (b_1, \ldots, b_t), (b_{t+1}, \ldots, b_{2t}), \ldots, (b_{N-t+1}, \ldots, b_N)$, where N stands for the length of D. It takes any distinct values $x_1, x_2, \ldots, x_n$ with $n > t-1$ and computes the shares as (x_i, y_i), $1 \le i \le n$, where y_i results from

$$\begin{bmatrix} x_1^0 & x_1^1 & \ldots & x_1^{t-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_i^0 & x_i^1 & \ldots & x_i^{t-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_n^0 & x_n^1 & \ldots & x_n^{t-1} \end{bmatrix} \begin{bmatrix} b_1 & \ldots & b_{N-t+1} \\ \vdots & \vdots & \vdots \\ b_t & \ldots & b_N \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_t \end{bmatrix}.$$

For reconstructing the document, a system of linear equations has to be solved.

In our architecture, we adopt the approach proposed by Krawczyk in [20]. This method represents a combination of both presented algorithms: the space-efficient (but only computationally secure) information dispersal algorithm of Rabin for a symmetrically encrypted document, and the perfectly secure (but not space-efficient) secret-sharing scheme of Shamir for the encryption key. When compared to a document share of Shamir having the length of the document itself, the length of Krawczyk share is usually much smaller because it is obtained as a pair of a document share and an encryption key share, and thus reduced to the length of the document divided by the threshold plus the length of the encryption key. The only drawback to be stated is that Krawczyk's method relies on the security of the symmetric encryption function.

This approach is in general based on the assumption that shareholders return correct shares. When this does not hold, i.e., some shares may be modified by possibly malicious parties, Krawczyk additionally guarantees authenticity of the document by using distributed fingerprints introduced in [60]. Here, the string of the hashes of shares is additionally being split by Rabin's information dispersal algorithm. The correctness of shares is then made sure by comparison of the corresponding hashes. In our architecture, we improve this approach by using two HMACs per share instead, see Section 6.

5.2.4. Privacy-preserving share identifiers

In order to ensure that the search key for shares stored in the clouds does not reveal any information about the medical record and the patient it relates to, we consider mechanisms based on cryptographic hash functions h. In our prototype we used SHA-1 for h [61]. The exact set of internal identifiers necessary to uniquely identify a MR depends on the concrete use case, but should be common knowledge and practice among all participating organizations. For example, using a unique internal patient identifier P-ID, a unique identifier HC-ID for health centers, and a share ID as inputs, the proxy at the HC where data should be stored or retrieved calculates the cryptographic hash of the concatenated internal identifiers: h(*P-ID, HC-ID, Share-ID*). This hash value does not provide any sensitive information about the patient identity to an external party or a CP, since it is nearly impossible to invert the hash function. With this procedure, we are addressing the requirements of medical record anonymity, unlinkability between medical records and patients, and medical record existence confidentiality (see Section 4).

As a countermeasure against dictionary attacks, the actual convention would be to execute the hash function h iteratively w times (e.g., $w=1000$ times, $h^{1000}(\ldots)$), for a fixed parameter w known among all partners. As our experiments show, this procedure has nearly no impact on latency for a single calculation, but would massively increase the effort to construct a dictionary of hash values for all possible input values. In future work, we will also investigate random salts and their management in cooperative settings.

5.3. Discussion

In this section, we critically reflect how and to what extent our architecture is able to satisfy the requirements specified in Section 4. An overview is given in Table 2. In general, it can be stated that most requirements can be satisfied by our current architecture, and to a high or at least medium degree. In the following, we focus on selected requirements and limitations of our current work, indicating ideas for future technical or organizational improvements.

Access right revocation: Revoking access rights for users on data stored in the clouds can be considered a challenge since revocable ABE is still an open research area [14]. Even if an easily revocable scheme would be implemented in our architecture, it is still necessary to update the shares of the secret-sharing layer as well. But if revocation is a relatively rare event, we think that the corresponding overhead is acceptable. An alternative approach to changing the ABE ciphertext and corresponding shares would be a refined local management and access control for ABE keys. If the local infrastructure is trustworthy, it could conduct the ABE encryption and decryption as an intermediary on behalf of the users, possibly revoking access to these operations once the role of a user changes. However, such an indirect key access needs further security analysis.

Table 2
Security requirements and corresponding measures (for # cf. Table 1).

Requirement	Main measures (#)	Qualitative assessment of fulfillment (high, medium, low, none)
Confidentiality of medical record content	1, 2, 4, 5, 6, 8, (9)	High
Confidentiality of medical record existence	4, 6, 8, 9	Medium
Anonymity of patient in a medical record	6, 8, 9	Medium
Unlinkability between medical records and patients	6, 8, 9	Medium
Integrity and authenticity of medical records	1, 2, 5, 6, 7, (8)	High
Availability of medical records	8, 10	High
Authentication of users and services	1, 3	High
Fine-grained authorization and access control	3, 4, 5, 6	High
Anonymity of user identity	Requests to CPs use organizational credentials	Medium
Auditing capability	Logging facility in client and proxy	Medium
Confidentiality of user access privileges	Requests to CPs use organizational credentials	Medium
Access right revocation	4, 5, 6 with republishing, see discussion	Low
Emergency exceptions	4, 5, 6 with special access policies	Low
Scalability of security measures	3, 5; and see discussion	Medium
Availability of systems	Multi-cloud approach; proxy redundancy	Medium
Archiving of medical records	See discussion	Medium
Efficiency	See discussion and experiments in Section 6	High
Usability	Needs further tool support	Medium

Emergency exceptions: The overriding of access control in specific and urgent cases is an open issue in our approach. Even in emergency cases, internal document identifiers need to be known and enough shares of relevant documents must be retrieved. In order to cope with the ABE encryption, ideas we are considering include the use on an emergency role that would have access to all encrypted attributes. However, secure key management for this role, in particular on mobile wireless devices prevalent in some emergency settings, will be challenging. We conclude that currently our architecture is more suitable for regular information sharing use cases in health care and yet not well-suited for emergency cases where client devices need an ad-hoc and mobile data access.

ABE key authority: In CP-ABE, the owner of the master key is able to generate private keys based on public parameters and an arbitrary set of attributes. The master key must be properly secured by both technological and organizational measures. A trusted key authority can be established, which assures that users can only request private keys for attributes that indeed pertain to them. Otherwise, a user could request private keys on improper attributes that could possibly satisfy an access structure. However, such a key authority involves a possible impediment for scaling securely to very large sets of clients and may be difficult to design redundantly.

Attribute verification should be enforced by an organizational process where a user or client system receives a signed data structure after successful attribute acknowledgment. This signed structure obtained can then be verified by a key authority upon private key requests and generation. Such a verification process could be realized as part of the Coordination Services on a separate infrastructure, similar to the certificate authority used to build the client signature certificate chain. It remains a challenge for future research to design the CP-ABE key authority in such a way that a private key for a unique attribute set is only issued once to the dedicated recipient and to mitigate the risk of improper key generation.

Scalability and cloud-based MCP: The Multi-Cloud Proxy (MCP) acts as an intermediary between authorized clients and multiple cloud data stores (CDS). As it allows to add connectors to attach specific storage locations into the system, each CDS must be configured with access credentials, such as security tokens or public-private key pairs [62], so that the MCP can prove authorization towards a CP and be granted access. In our extended architecture (Fig. 2), we contemplate instances of the MCPs to be operated in a private Service Cloud being accessible over the Internet, in order to increase overall scalability and proxy availability, as well as to support mobile devices with low computing power. Such a deployment scenario entails greater exposure to attacks, thus CDS credentials must be properly protected. One way to overcome this limitation could be to asymmetrically encrypt all credentials and configuration data by a public key to allow for persistency. The decryption process then could use, for example, a password-derived private key that must be entered by a security officer during the system boot procedure of a MCP.

6. Evaluation: implementation and experiments

6.1. Multi-cloud proxy

As the most important component of our architecture presented in Section 5, we designed and implemented a Multi-Cloud Proxy that acts as an intermediary between clients at a HC and Cloud Data Stores (CDS) operated by CPs. Its objective is to provide authenticated and authorized clients with a secure storage facility, which also involves secret-sharing techniques. Clients can authenticate and authorize themselves against the proxy, for example by providing client certificates whose chain is fully verifiable up to the root certificate authority of the architecture.

The proxy offers a REST [63] service interface and accepts incoming data such as an MR. It applies secret

sharing by Krawczyk [20] as described in Section 5.2.3 to produce a configurable amount of shares that are distributed to multiple registered CDS. All Privacy-Preserving Share Identifiers are computed by repeatedly applying the SHA-1 cryptographic hash function (see Section 5.2.4) on the client's input of identification parameters and the corresponding share number being processed.

For each storage request, the Krawczyk procedure is applied: A new, randomly generated encryption key is used to encrypt the data before applying Rabin's Information Dispersal Algorithm to the ciphertext and creating shares of the encryption key by using Shamir's secret sharing. As current cipher, we use AES [64] with a 256 bit key in Cipher Block Chaining (CBC) mode to allow for parallelized decryption and prevent plain-text patterns and reoccurring plain-text blocks from being identifiable in the ciphertext [65]. Both the cipher-algorithm identifier and initialization vector get encoded alongside the encryption key before splitting them into Shamir key shares. This allows to generate a new random key for each storage request and gives the possibility to dynamically switch between ciphers at system runtime.

In order to guarantee data integrity, the MCP computes two cryptographic Hash-based Message Authentication Codes (HMAC) for each share before distributing them to the CDS. Authentication codes at share level give us the ability to detect which data stores and communication channels have been tampered. Our current implementation uses HMAC-SHA-1 [66] on all shares, based on the encryption key being used in Krawczyk share creation. This allows authorized Multi-Cloud Proxies of other health centers to verify share integrity, as they are able to recover the decryption key once threshold t many shares have been retrieved. Because this leaves the chance of the HMAC being compromised if an attacker has gained access to the t data stores, we apply an additional HMAC using a secret key that is internally protected in the key store of the proxy. With this in place, the share-creating proxy can detect tampering even if the Krawczyk encryption key was compromised. We note that even in case of such a large security breach, the integrity of the MR is still protected by the digital signature, and its confidentiality by ABE.

For end-to-end integrity and authenticity during the entire roundtrip of data transfer, i.e., from the client via the MCP to shares being stored in the CDSs, and later back to (the same or) another client, we use digital signatures created at the HC's client software to sign and verify all content that is stored in the clouds. Each client generates the signature using the DSA algorithm [67]. We also consider adding X.509 certificates to the signature process in future, so that other authorized clients are able to successfully verify signatures during cross-HC requests without storing any public keys beforehand.

The Multi-Cloud Proxy both acts as a web-service server and client. Towards health centers, it provides a REST web service interface that allows authenticated clients to store a MR as encrypted shares in the clouds (*PUT*), to have the proxy recover them (*GET*), and to perform updates and deletions on previously stored secret objects. The MCP then acts as a web-service client that uses e.g., REST or SOAP web services provided by CDSs. We designed the MCP to be easily extensible in order to connect to a great variety of CDSs and allow for future extensibility. The MCP exposes an interface definition for cloud connectors for which specific implementations can be developed. Currently, we provide support for Amazon AWS S3 [68] and Google Cloud Storage [69]. Currently, our MCP supports the entire workflow as depicted in Fig. B1 for *PUT* requests, including all inverse operations involved in *GET* requests.

The connector design also provides a facility to integrate and utilize local network storages devices such as SANs, FTP or SFTP servers, which are only accessible within the local network of the MCP. These could function as a local persistence layer for data shares produced by the MCP (alongside storage into cloud data stores). From a security perspective, such a configuration can be used to distribute less than the threshold amount of shares into the CDSs, making an MR unrecoverable for an attacker even if he would gain control over all cloud data stores.

Composition and dependencies: We implemented our MCP as a Java Web Application that runs in a servlet container and allows to publish its web services towards client systems. Building it as a web application also allows to deploy and operate the MCP in a Service Cloud using a Platform-as-a-Service approach. All communication channels use Transport-Layer Security (TLS). In order to avoid data remanence, all operations such as share processing and secret recovery are performed in-memory, without the need for disk I/O.

Secret-sharing implementations of Shamir, Rabin's Information Dispersal Algorithm, and Krawczyk scheme have all been newly implemented without referring to any third-party libraries. For web services, the MCP uses the JAX-RS (JSR 311) reference implementation [70] for providing REST services to clients, and uses Apache HTTP Client [71] for our current CDS connectors to communicate with Amazon AWS S3 and Google Cloud Storage. All available data-store connectors are pluggable and loaded at runtime, thus the CDS configuration can be adjusted at any time.

The cryptographic operations that are involved in the Krawczyk scheme, such as encrypting data before producing IDA shares or for computing secure hash digests and HMACs, all use the crypto engines of Bouncy Castle [72] as JCE provider. Cipher-Policy Attribute Based Encryption (CP-ABE) is currently performed by a third party library [73], which uses the Pairing-Based Cryptography library jPBC [74].

Our implementation of IDA for both share creation and secret recovery, HMAC creation and verification as well as cloud *PUT* and *GET* operations are executed using multi-threading.

6.2. Experiments

In our architecture, the MCP may be accessed from a multitude of clients within a HC, therefore its performance is critical, especially for allowing fast access to previously stored records. Although we consider applying load-balancing techniques in the future to counteract peak-usage times and mitigate single points of failure, we conducted experiments on the processing speed using only one MCP, in order to

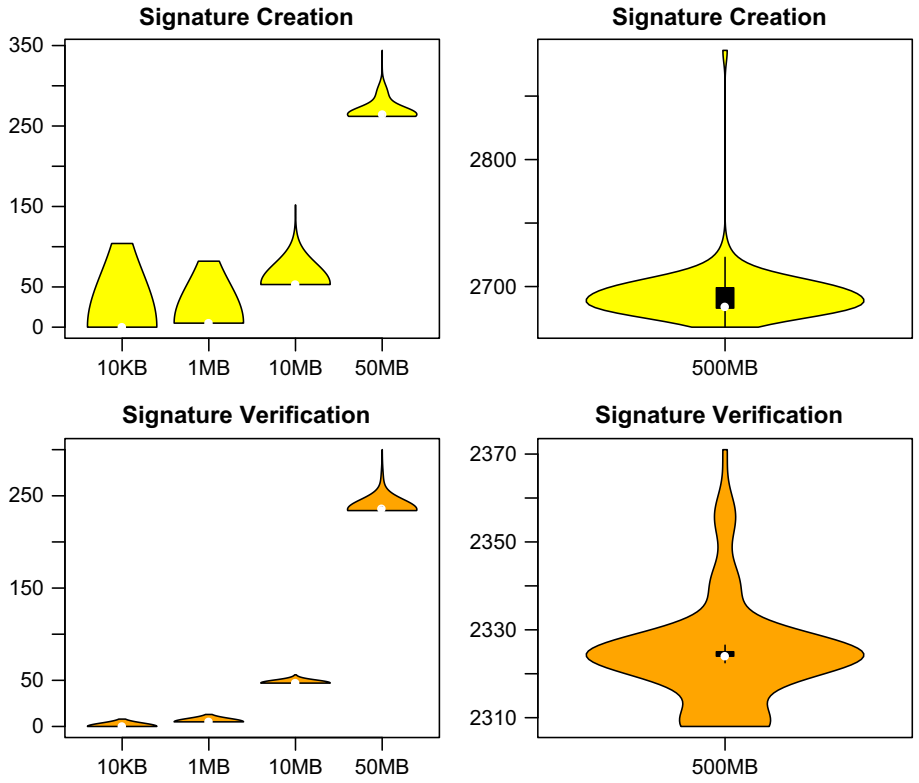

Fig. 4. Execution times of signature creation (*ms*).

clearly show its performance for each sub process of the workflow without such techniques.

We performed experiments to benchmark the entire communication round-trip as shown in Fig. B1, including the inverse process. Starting with a user who initiates a storage request to the client software, which includes a confidential MR together with its internal identifier (see Section 5.2.4), the client software performs CP-ABE encryption according to a specified access structure that is isomorphic to the second example given in Section 5.2.2. The client also digitally signs the ciphertext data before passing it to the MCP within a REST request. Since both the client and the MCP are likely to be deployed within the same network (in the *Basic Architecture*), we used the same machine for benchmarking signature creation, verification as well as CP-ABE encryption and decryption. Transfer times between client and MCP depend on the local network setup of a deployment, but we consider them to be low in general and therefore do not reflect them within the "overall process" benchmarks of our experiments.

Once the data has been transmitted to the MCP, we measure the execution time to create Krawczyk shares (indicated as "Share Creation" in Table A2), which in our system consists of: (i) generating a random secret key of 256 bit, (ii) encrypting the data with this key using AES, (iii) creating shares of the encrypted data using Rabin's IDA, (iv) creating shares of the encoded secret key including its algorithm parameters and initialization vector using Shamir secret sharing and (v) computing two HMAC-SHA-1 on the all shares. The "Secret Recovery" benchmark includes (vi) tampering detection by two HMAC verifications on all shares, (vii) secret recovery of the Krawczyk encryption key using Shamir secret sharing, (viii) recovery of the data ciphertext using IDA and (ix) symmetric decryption using the recovered secret key.

Medical Records can vary in size depending on the data contained, which can be plain text or consists of photographic images or even movies [7]. Our sample documents therefore cover binary lengths of 10 KB, 1 MB, 10 MB, 50 MB and 500 MB. No compression was applied before ABE, thus the encrypted records are of equal or greater size when arriving at the MCP to be split into shares. It should also be noted that the share-creation process produces n shares, however, its inverse process only uses t shares necessary for secret recovery.

The first set of experiments executes all processes described above and uses a (3,4)-Krawczyk scheme for secret sharing, a high level summary is shown in Figs. 4–7. These figures show "violin plots" [75], which are combining box plots and kernel density plots [76]. Detailed results are given in Table A2. All results are based on 1000 repetitions, except for 500 MB being repeated 500 times. Both client and MCP were operated on a Windows 7 Profession 64 bit machine within Oracle JRE 1.6.0_39 and Java HotSpot 64-bit Server VM. The machine uses an Intel Core i5 2500 K that runs on 4x4841 MHz with 8 GBytes of DDR3 RAM operating in Dual Channel on 686,9 MHz (Fig. 8).

Our second set of experiments takes an already ABE-encrypted and signed input document from a client, but in

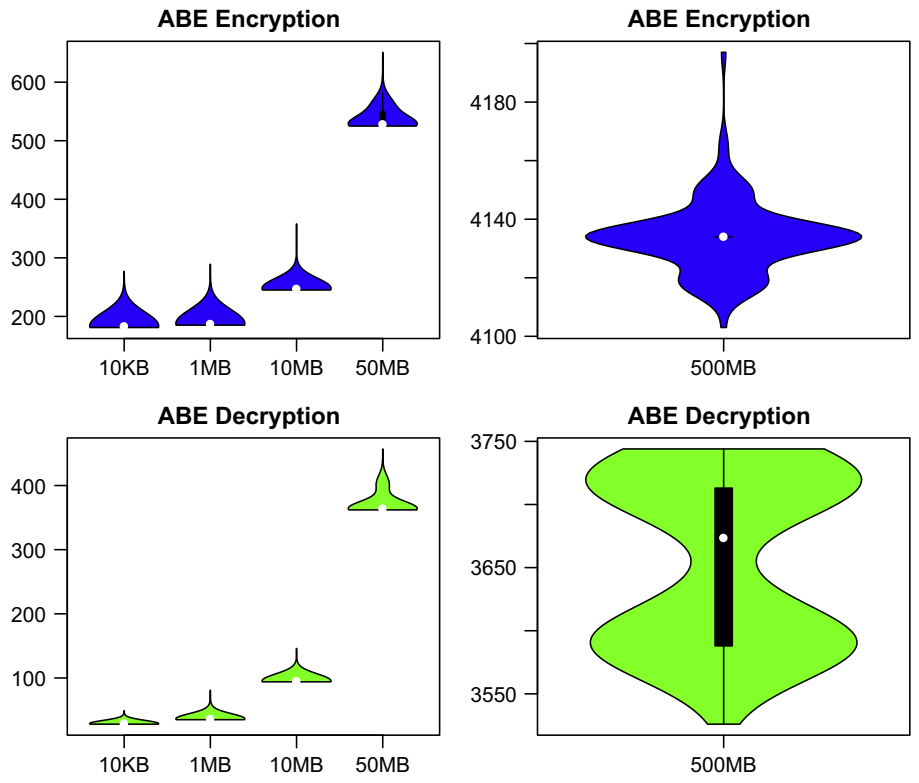

Fig. 5. Execution times of ABE encryption and decryption (*ms*).

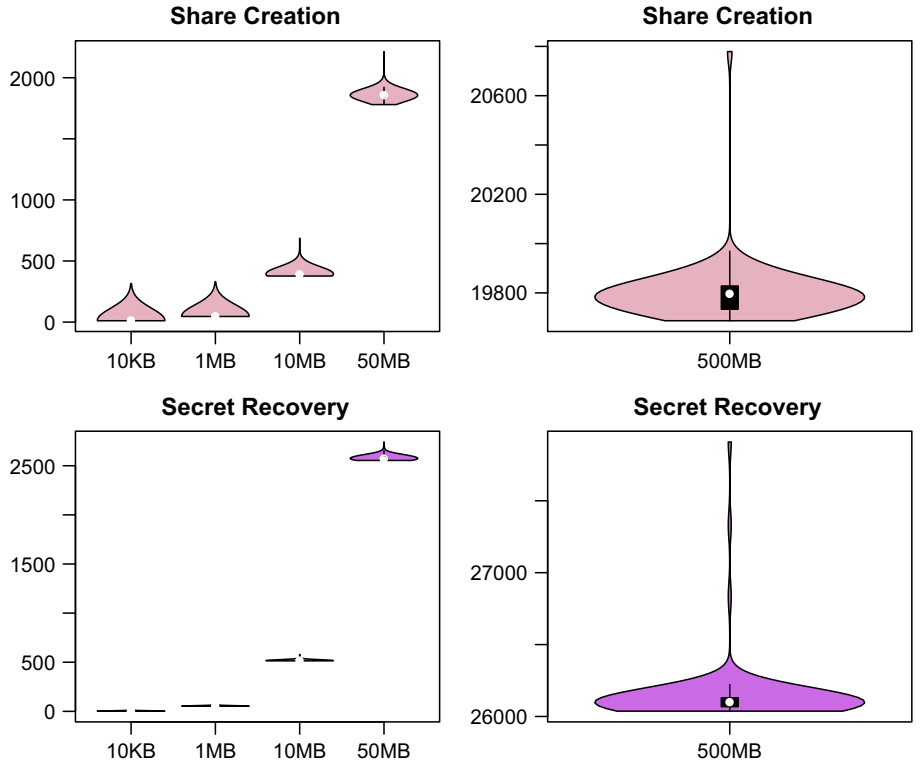

Fig. 6. Execution times of Krawczyk (3,4)-scheme (*ms*).

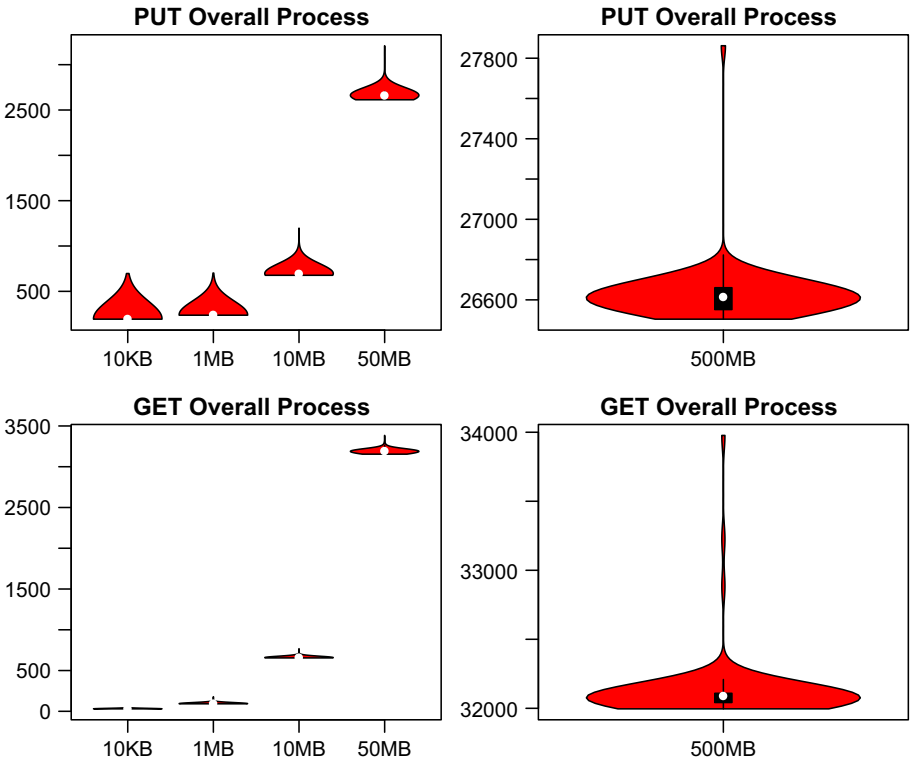

Fig. 7. Overall time of *PUT* and *GET* without cloud access (*ms*).

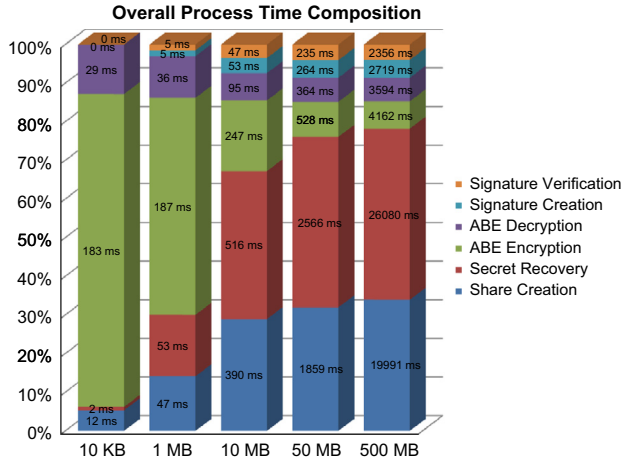

Fig. 8. Time composition of the overall process (without cloud access).

contrast to the first experiment, shows execution times for share creation and secret recovery using a (8,10)-Krawczyk scheme (see Table A3 and Fig. A1). The rationale was to investigate the performance when more shares are generated with a higher threshold, which would increase the efforts of an adversary to gather enough shares for reconstruction.

The third set of experiments measures transfer times between the MCP and four CDSs. These four CDSs, which must be run by independent providers in any real-world implementation, are emulated in our experiments by globally different locations of only two independent CPs: two buckets of Amazon AWS S3 [68], and further two buckets at Google Cloud Storage [69]. Here, one bucket of Amazon was located at a data farm in Ireland and the remaining three in the U.S. We note that even in this experimental constellation, our (3,4)-Krawczyk scheme would prevent any single CP from reconstructing the MR. For this experiment, the MCP was running in a virtual machine physically hosted in a university network in Germany. The virtual machine was running Ubuntu 10.04.4 LTS with a virtual 100 Mbit/s connection. This third set of experiments is based on 500 repetitions per data size, except for 500 MB, where the experiments were

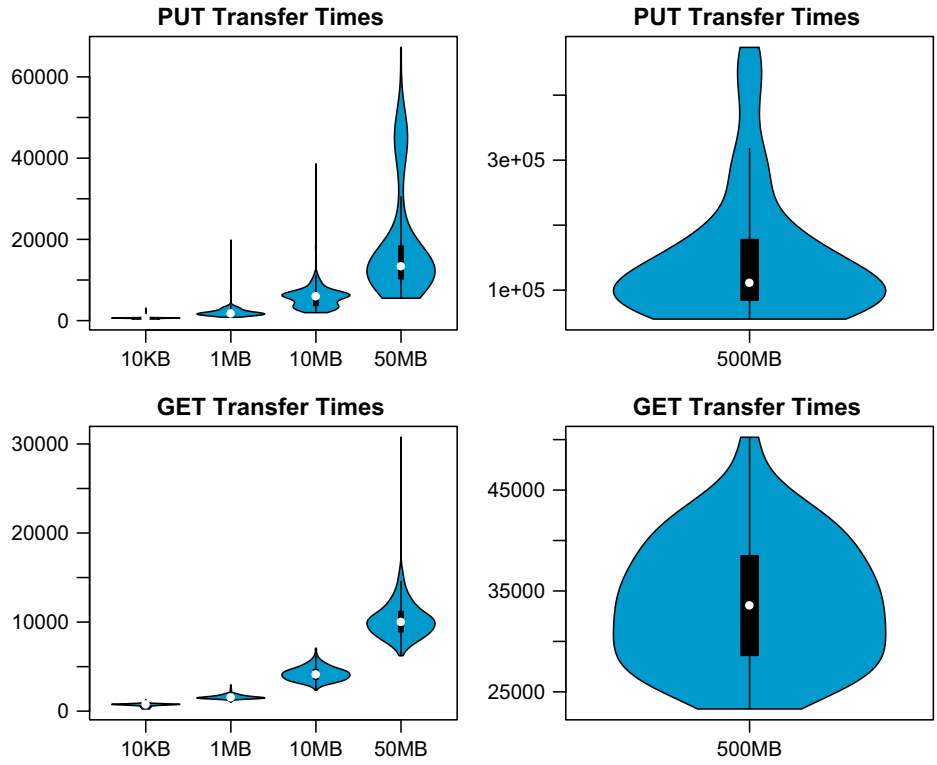

Fig. 9. Transfer times of (3,4)-shares to/from testing clouds (*ms*).

repeated 100 times. Since these results vary a lot depending on the choice, location, and configuration of the cloud servers used, we give them separately in Fig. 9, see also Table A1 for detailed results.

Our experiments indicate that all processing steps of our architecture can be accomplished with good performance. When comparing execution times against input document size, the computational overhead is highest for the smallest samples of 10 KB. For all other document sizes, data throughput is almost constant for share creation, secret recovery, signature creation and verification. ABE involves an initial computational overhead leading to higher throughput on larger documents. While secret sharing and recovery only makes up 6.1% of the overall process time of a 10 KB document, it accounts for 78.2% of the time on a 500 MB document.

From a usability perspective, the execution time of *PUT* requests is relatively uncritical because pending jobs can be queued. However, it is more important for *GET* requests as a user's waiting time should be minimal. Bandwidth and network connectivity are the major limitation in this respect, however this applies for large files in general and is not specific to our architecture. Another major bottleneck, however, is CPU processing power and memory resources since the current implementation performs all operations in-memory. This requirement is particularly important to avoid data-remanence if the MCP was running in an external service cloud. When deploying the MCP on a mobile device (see Fig. 2), disk I/O could become necessary and would increase execution time. It is therefore favorable to operate the MCP in a fast cloud-server environment, allowing clients to delegate computational cost. In future, we intent to add load-balancing and perform stress tests to simulate operation within a larger organization and multiple concurrent client requests.

7. Limitations and future work

In future work, we will aim to address several open tasks. First of all, for large deployment scenarios, the problem of scalable Cooperation Services and in particular key management needs to be solved. Management and tool support for inter-organizational access-control policies is an important area of future research, as well as providing software for a mapping from RBAC models to ABE access structures in order to enhance usability. A separate important research line could be the integration of new components that offer anonymization of patient data by following concepts such as k-anonymity or differential privacy.

In our current architecture, potential single points of security failure such as the ABE key authority should become decentralized if possible, and separation of security duties within Cooperation Services established. Concerning the Multi-Cloud Proxy, we aim for supporting more interfaces to different cloud providers and support for streaming data and load-balancing. Moreover, we aim to add P2P interfaces for different Distributed Hash Tables and BitTorrent. In the area of secret sharing, recent advances such as mobile proactive secret sharing [77] need to be investigated. For construction of external identifiers, managing salts

against dictionary attacks could provide interesting inter-organizational challenges.

A stronger involvement of the patient into our architecture is also considered for future work. Finally, pilot tests and usability studies, for example in the context of healthcare projects such as the one mentioned in the introduction, will provide interesting extensions of the current work.

8. Conclusion

Currently, there are many security and privacy challenges impeding the wide adoption of cloud computing in the healthcare domain. In this paper, we presented a novel architecture for inter-organizational data sharing and its implementation, which provides a high level of security and privacy for patient data in semi-trusted cloud computing environments. This architecture features attribute-based encryption for selectively authorizing access to data and cryptographic secret-sharing in order to securely distribute data across multiple clouds, reducing the adversarial capabilities of curious cloud providers. Our implementation and evaluation by several experiments indicated the practical feasibility and good performance. Future work will address inter-organizational aspects of key management and RBAC policy management, usability studies, and several enhancements for the Multi-Cloud Proxy.

Appendix A. Detailed experimental results

See Fig. A1 and Tables A1–A3.

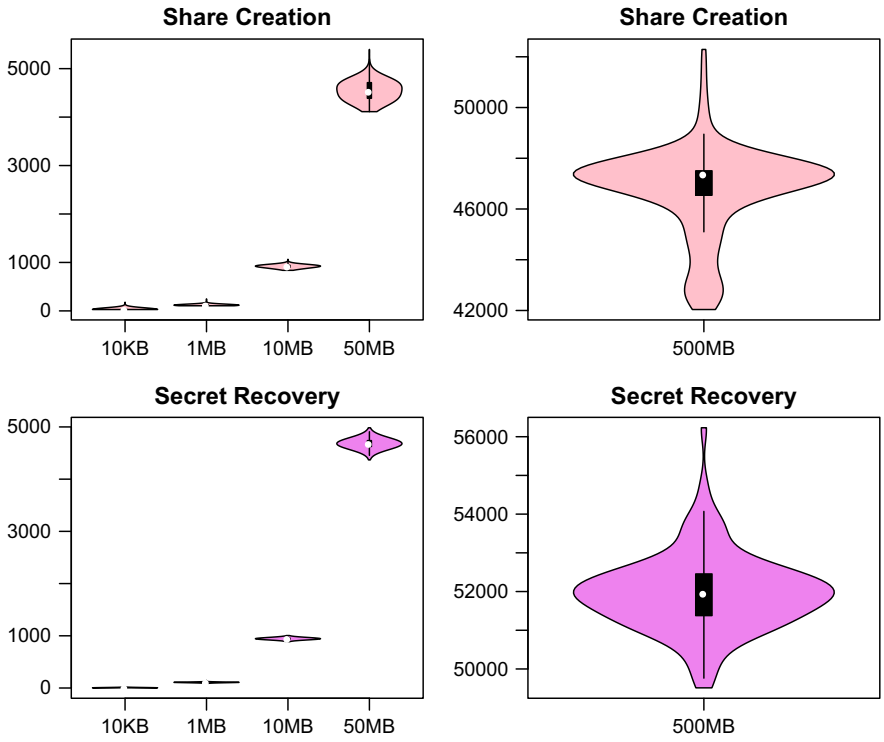

Fig. A1. Execution Times of Krawczyk (8,10)-Scheme (*ms*).

Table A1
Transfer times of (3,4)-shares to/from a multi-cloud (*ms*).

Operation	Size	Mean	Lower quartil	Median	Upper quartil	Std. deviation
PUT Request	10 KB	603.54	482	630	646	195.67
	1 MB	2040.30	1442	1768	2330	1316.72
	10 MB	5671.70	3760	5994	6730	2687.63
	50 MB	18,626.39	10239	13,381	18,517	13,240.29
	500 MB	148,947.69	83,435	111,205	179,774	99,954.70
GET Request	10 KB	696.27	654	749	785	163.70
	1 MB	1596.03	1443	1541	1709	236.67
	10 MB	4181.62	3612	4121	4659	755.31
	50 MB	10,222.25	8922	10010	11,192	2033.59
	500 MB	33,892.94	28,570	33,563	38,575	6113.96

Table A2
Execution times of Krawczyk (3,4)-scheme (*ms*).

Operation	Size	Mean	Lower quartil	Median	Upper quartil	Std. deviation
Share creation	10 KB	12.47	12	12	12	9.66
	1 MB	48.03	47	47	48	9.10
	10 MB	393.73	389	390	396	11.54
	50 MB	1858.16	1837	1859	1871	24.95
	500 MB	20,068.33	19,923	19,991	20,266	189.64
Secret recovery	10 KB	2.07	2	2	2	0.56
	1 MB	53.22	53	53	53	1.85
	10 MB	517.91	516	516	517	5.27
	50 MB	2579.92	2560	2566	2598	23.47
	500 MB	26,004.17	25,874	26,080	26,119	147.30
ABE Encryption	10 KB	183.39	183	183	184	3.19
	1 MB	187.52	187	187	188	3.78
	10 MB	248.93	247	247	248	5.65
	50 MB	537.32	527	528	549	16.26
	500 MB	4251.16	4142	4162	4363	110.83
ABE Decryption	10 KB	28.92	29	29	29	1.25
	1 MB	36.41	36	36	36	2.99
	10 MB	94.90	94	95	95	1.98
	50 MB	370.95	363	364	365	15.60
	500 MB	3594.86	3586	3594	3601	14.43
Signature creation	10 KB	0.10	0	0	0	3.29
	1 MB	5.09	5	5	5	2.44
	10 MB	53.52	53	53	54	3.27
	50 MB	268.29	264	264	265	10.09
	500 MB	2721.37	2718	2719	2722	18.40
Signature verification	10 KB	0.01	0	0	0	0.26
	1 MB	5.02	5	5	5	0.27
	10 MB	47.07	47	47	47	0.38
	50 MB	236.30	234	235	236	6.56
	500 MB	2357.54	2355	2356	2358	5.16
Overall process time	10 KB	229.29	228	228	229	16.98
	1 MB	337.18	335	336	337	17.78
	10 MB	1358.62	1351	1356	1363	21.35
	50 MB	5853.68	5838	5853	5866	33.94
	500 MB	59,000.25	58,857	58,941	59,181	200.57

Table A3
Execution times of Krawczyk (8,10)-Scheme (*ms*).

Operation	Size	Mean	Lower quartil	Median	Upper quartil	Std. deviation
Share creation	10 KB	29.65	29	29	30	5.05
	1 MB	118.65	115	118	123	6.36
	10 MB	917.91	899	921	941	30.65
	50 MB	4536.79	4388	4529	4712	179.07
	500 MB	46,725.89	46,526	47,370	47,507	1753.89
Secret recovery	10 KB	4.12	4	4	4	0.82
	1 MB	107.64	106	109	110	4.26
	10 MB	945.21	932	945	957	18.16
	50 MB	4680.55	4625	4679	4738	85.35
	500 MB	51,965.58	51,370	51,951	52,462	998.70

Appendix B. Communication flow

See Fig. B1.

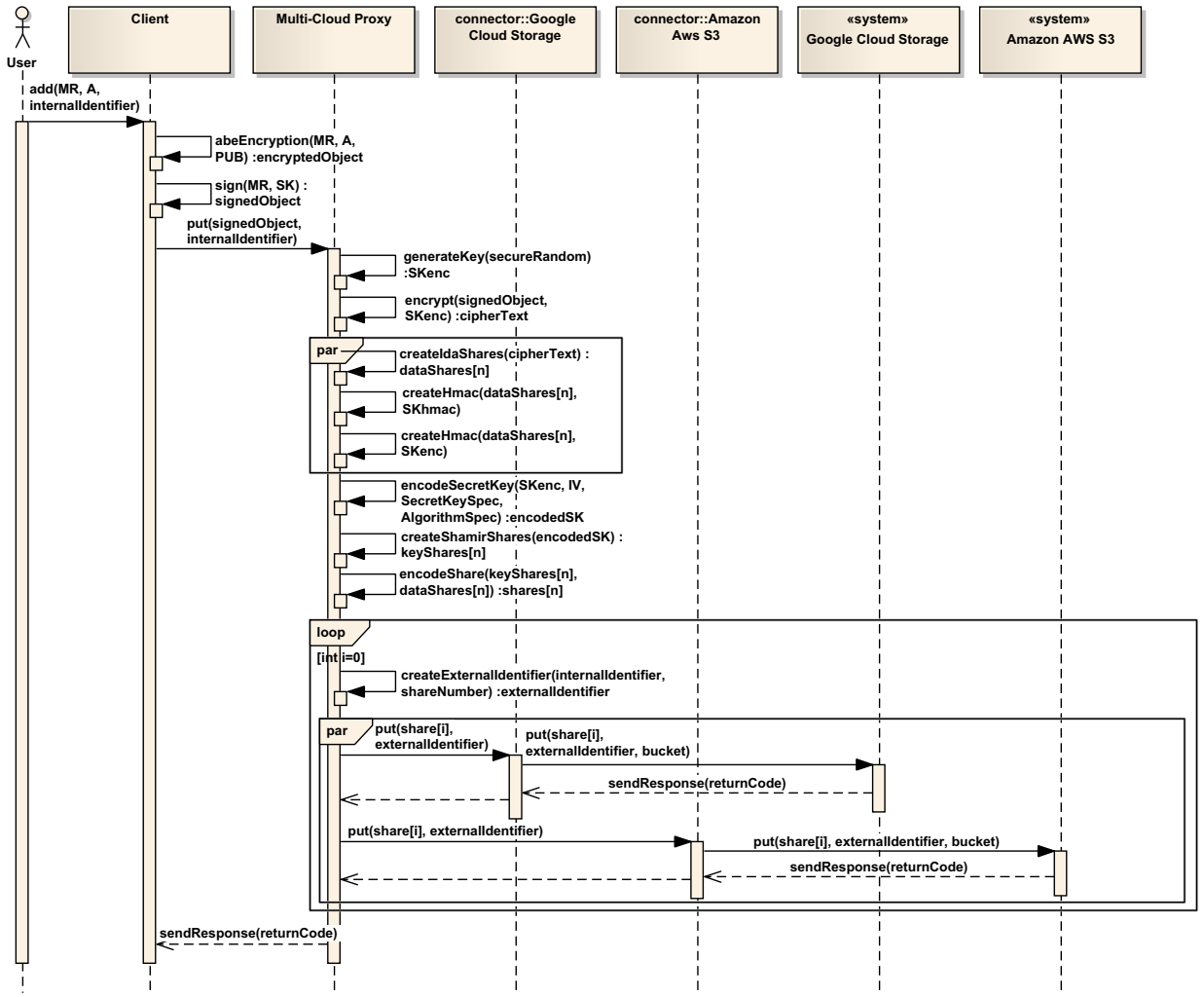

Fig. B1. Workflow and communication overview of a *PUT* request.

References

[1] R.F. Chong, Changing the world: big data and the cloud, ⟨http://www.theatlantic.com/sponsored/ibm-cloud-rescue/archive/2012/09/changing-the-world-big-data-and-the-cloud/262065/⟩ (2012).
[2] V. Koufi, F. Malamateniou, G. Vassilacopoulos, Ubiquitous access to cloud emergency medical services, in: Proceedings of 10th IEEE International Conference on Information Technology and Applications Biomedicine, 2010.
[3] C.O. Rolim, F.L. Koch, C.B. Westphall, J. Werner, A. Fracalossi, G.S. Salvador, A cloud computing solution for patient's data collection in health care institutions, in: Proceedings of the 2nd Telemedicine and Social Medicine, 2010.
[4] G. Kanagaraj, A. Sumathi, Proposal of an open-source cloud computing system for exchanging medical images of a hospital information system, in: Proceedings of 3rd International Conference on Trends in Information Sciences and Computing, 2011.
[5] M. Poulymenopoulou, F. Malamateniou, G. Vassilacopoulos, E-EPR: a cloud-based architecture of an electronic emergency patient record, in: Proceedings of Pervasive Technologies Related to Assistive Environments, 2011.
[6] N. Karthikeyan, R. Sukanesh, Cloud based emergency health care information service in India, J. Med. Syst. 6 (6).
[7] OsiriX DICOM Viewer, DICOM Sample Image Sets, ⟨http://www.osirix-viewer.com/datasets/⟩, 2013.
[8] P. Mell, T. Grance, The NIST Definition of Cloud Computing, Technical Report, National Institute of Standards and Technology, 2012.
[9] S. Shini, T. Thomas, K. Chithraranjan, Cloud based medical image exchange-security challenges, in: Proceedings of International Conference on Modelling, Optimization and Computing, 2012.
[10] K.A. Ratnam, D.D. Dominic, Cloud services—enhancing the Malaysian Healthcare Sector, in: Proceedings of International Conference on Computer and Information Science, 2012.
[11] S. Basu, A. Karp, J. Li, J. Pruyne, J. Rolia, S. Singhal, J. Suermondt, R. Swaminathan, Fusion: managing healthcare records at cloud scale,

in: IEEE Computer, Special Issue on Move Toward Electronic Health Records.
[12] L. Guo, F. Chen, L. Chen, X. Tang, The building of cloud computing environment for E-health, in: Proceedings of International Conference on E-Health Networking, Digital Ecosyst. and Technologies, 2010.
[13] M. Li, S. Yu, N. Cao, W. Lou, Authorized private keyword search over encrypted personal health records in cloud computing, in: Proceedings of 31st International Conference on Distributor Computer Systems, 2011.
[14] M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou, Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, IEEE Trans. Parallel Distr. Syst. 24 (1) (2012) 131–143.
[15] T.-S. Chen, C.-H. Liu, T.-L. Chen, C.-S. Chen, J.-G. Bau, T.-C. Lin, Secure dynamic access control scheme of PHR in cloud computing, J. Med. Syst. 6 (6).
[16] L. Chen, D.B. Hoang, Novel data protection model in healthcare cloud, in: Proceedings of IEEE International Conference on High Performance Computing and Communications, 2011.
[17] M. Li, S. Yu, K. Ren, W. Lou, Securing personal health records in cloud computing: patient-centric and fine-grained data access control in multi-owner settings, in: Proceedings of SecureComm 2010, LNICST, vol. 50, 2010, pp. 89–106.
[18] J.A. Akinyele, C.U. Lehmann, M.D. Green, M.W. Pagano, Z.N.J. Peterson, A.D. Rubin, Self-protecting Electronic Medical Records Using Attribute-based Encryption, Cryptology ePrint Archive, Report 2010/565, 2010.
[19] U.S. Department of Health & Human Services, Health Information Privacy, ⟨http://www.hhs.gov/ocr/privacy/⟩, 2013.
[20] H. Krawczyk, Secret sharing made short, in: Proceedings 13th Annual International Cryptology Conference on Advances in Cryptology, 1994, pp. 136–146.
[21] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: Proceedings of IEEE Symposium on Security and Privacy, 2007, pp. 321–334.
[22] T. Ermakova, B. Fabian, In submission: secret sharing for health data in multi-provider clouds, in: Proceedings of IEEE Conference on Business Informatics, 2013.
[23] A. Hevner, S. Chatterjee, Design Research in Information Systems: Theory and Practice, Springer, New York, 2010.
[24] K. Peffers, T. Tuunanen, M. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, J. Manage. Inf. Syst. 24 (3) (Winter 2007–2008) 45–77.
[25] D.B. Hoang, L. Chen, Mobile Cloud for Assistive Healthcare (MoCAsH), in: Proceedings of IEEE Asia-Pacific Services Computing Conference, 2010.
[26] S. Sharieh, F. Franek, A. Ferworn, Using cloud computing for medical applications, in: Proceedings of 15th Communications and Networking Simulation Symposium, 2012.
[27] R.-D. Berndt, M.C. Takenga, S. Kuehn, P. Preik, G. Sommer, S. Berndt, SaaS—platform for mobile health application, in: Proceedings of 9th International Multi-Conference on Systems Signals and Devices, 2012.
[28] M. Deng, M. Nalin, M. Petkovié, I. Baroni, A. Marco, Towards trustworthy health platform cloud, in: Proceedings of IEEE 4th International Conference on Cloud Computing, 2012.
[29] OMG, Business Process Model and Notation, ⟨http://www.bpmn.org/⟩, (2013).
[30] TRESOR Project, ⟨http://www.cloud-tresor.com⟩ (2013).
[31] C. He, X. Jin, Z. Zhao, T. Xiang, A cloud computing solution for hospital information system, in: Proceedings of IEEE International Conference on Intelligent Computing and Intelligent System, 2010.
[32] Q. Huang, L. Ye, M. Yu, F. Wu, R. Liang, Medical information integration based cloud computing, in: Proceedings of International Conference on Network Computing and Information Security, 2011.
[33] L. Hardesty, Big medical data, ⟨http://web.mit.edu/newsoffice/2013/big-medical-data-0125.html⟩, January 25, 2013.
[34] Cloud4Health Project, ⟨http://www.cloud4health.de/⟩.
[35] GeneCloud Project, ⟨http://transinsight.com/genecloud-2/?lang=en⟩, 2013.
[36] D. Vazhenin, Cloud-based web-service for health 2.0., in: Proc. Joint Int. Conf. on Human-Centered Computer Environments, 2012.
[37] ENISA, Cloud computing: benefits, risks and recommendations for information security, ⟨http://www.enisa.europa.eu/⟩, 2009.
[38] NIST, Guidelines on Security and Privacy in Public Cloud Computing, NIST Special Publication 800-144 (December 2011).
[39] A. Juels, A. Oprea, New approaches to security and availability for cloud data, Commun. ACM 56 (2) (2013) 64–73.
[40] T. Ermakova, B. Fabian, In submission: security and privacy requirements for adopting cloud computing in healthcare scenarios, in: Submission to Americas Conference on Information System, 2013.
[41] H. Loehr, A.-R. Sadeghi, M. Winandy, Securing the E-health cloud, in: Proceedings ACM International Health Informatics Symposium, 2010.
[42] I.M. Abbadi, M. Deng, M. Nalin, A. Martin, M. Petkovic, I. Baroni, Trustworthy middleware services in the cloud, in: Proceedings of 3rd International Workshop on Cloud Data Management, 2011.
[43] Z.-R. Li, E.-C. Chang, K.-H. Huang, F. Lai, A secure electronic medical record sharing mechanism in the cloud computing platform, in: Proceedings of IEEE 15th International Symposium on Consumer Electronics, 2011.
[44] Y.-Y. Chen, J.-C. Lu, J.-K. Jan, A secure EHR system based on hybrid clouds, J. Med. Syst. 5 (5).
[45] S. Yu, C. Wang, K. Ren, W. Lou, Achieving secure, scalable, and fine-grained data access control in cloud computing, in: Proceedings of 29th Conference on Information Communications, 2010.
[46] S. Narayan, M. Gagne, R. Safavi-Naini, Privacy preserving EHR system using attribute-based infrastructure, in: Proceedings of ACM Workshop on Cloud Computing Security Workshop, 2010.
[47] A. Bessani, M. Correia, B. Quaresma, F. Andre, P. Sousa, Depsky: dependable and secure storage in a cloud-of-clouds, in: Proceedings of 6th European Conference on Computer, 2011, pp. 31–46.
[48] B. Fabian, S. Gürses, M. Heisel, T. Santen, H. Schmidt, A comparison of security requirements engineering methods, Requir. Eng. 15 (1) (2010) 7–40.
[49] W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice Hall, Upper Saddle River, New Jersey, 2010.
[50] T. Dierks, E. Rescorla, The Transport Layer Security ({TLS}) Protocol, ⟨http://www.ietf.org/rfc/rfc4346.txt⟩, 2006.
[51] OASIS, Web Services Security v1.1.1, ⟨https://www.oasis-open.org/standards#wssv1.1.1⟩, 2012.
[52] S. Shim, G. Bhalla, V. Pendyala, Federated identity management, IEEE Comput. 38 (12) (2005) 120–122.
[53] D.F. Ferraiolo, D.R. Kuhn, R. Chandramouli, Role-Based Access Control, Artech House, Boston, London, 2007.
[54] M. Krawczyk, H. Bellare, R. Canettia, HMAC: keyed-hashing for message authentication, ⟨http://www.ietf.org/rfc/rfc2104.txt⟩, 1997.
[55] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, Proposed NIST standard for role-based access control, ACM Trans. Inf. Syst. Security 4 (3) (2001) 224–274.
[56] Benjamin Fabian, Steffen Kunz, Sebastian Müller, Oliver Günther, Secure federation of semantic information services, Decis. Support Syst. 55 (2013) 385–398, (special issue on "Data, information and analytics as services"). ⟨http://dx.doi.org/10.1016/j.dss.2012.05.049⟩.
[57] L. Ibraimi, M. Asim, M. Petkovic, Secure Management of Personal Health Records by Applying Attribute-Based Encryption, Technical Report, University of Twente, 2009.
[58] A. Shamir, How to share a secrets, Commun. ACM 22 (11) (1979) 612–613.
[59] M. Rabin, Efficient dispersal of information for security, load balancing, and fault tolerance, J. ACM 36 (1989) 335–348.
[60] H. Krawczyk, Distributed fingerprints and secure information dispersal, in: Proceedings of 12th Annual ACM Symposium on Principles of Distributed Computing, 1993, pp. 207–218.
[61] D. Eastlake, P. Jones, US Secure Hash Algorithm 1 (SHA1), ⟨http://www.ietf.org/rfc/rfc3174.txt⟩, 2001.
[62] Google Cloud Storage – Using Service Accounts for Authentication, ⟨https://developers.google.com/storage/docs/authentication#service_accounts⟩, 2013.
[63] R.T. Fielding, Chapter 5: Representational state transfer (REST), architectural styles and the design of network-based software architectures, Dissertation.
[64] NIST, Advanced Encryption Standard (AES), National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 197 (November 2001).
[65] NIST, Recommendation for Block Cipher Modes of Operation – Methods and Techniques, NIST Special Publication 800-38 A, 2001.
[66] NIST, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, 2008.
[67] NIST, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-3, 2009.
[68] Amazon Simple Storage Service (Amazon S3), ⟨http://aws.amazon.com/s3/⟩, 2013.
[69] Google Cloud Storage – Cloud Platform, ⟨https://cloud.google.com/products/cloud-storage⟩, 2013.
[70] Jersey, ⟨http://jersey.java.net/⟩, 2013.

[71] Apache HttpClient 4.2.3, ⟨http://hc.apache.org/httpcomponents-client-ga/index.html⟩, 2013.
[72] Bouncy Castle Crypto APIs, ⟨http://www.bouncycastle.org/java.html⟩, 2013.
[73] J. Wang, J. Perrochet, M. Grossi, S. Weiland, Java realization for "ciphertext-policy attribute based encryption" (CP-ABE), ⟨https://github.com/wakemecn/cpabe⟩, 2013.
[74] A. De Caro, The Java Pairing Based Cryptography Library (jPBC), ⟨http://gas.dia.unisa.it/projects/jpbc/⟩, 2012.
[75] ⟨http://cran.r-project.org/web/packages/vioplot/vioplot.pdf⟩, February 15, 2013.
[76] J.L. Hintze, R.D. Nelson, Violin plots: a box plot-density trace synergism, Am. Stat. 52 (2) (1998) 181–184.
[77] D.A. Schultz, B. Liskov, M. Liskov, MPSS: mobile proactive secret sharing, ACM Trans. Inf. Syst. Security 13 (4) 34).