I. Agudo, A. El Kaafarani, D. Nuñez, and S. Pearson, "A Technique for Enhanced Provision of Appropriate Access to Evidence across Service Provision Chains", *10th International IFIP Summer School on Privacy and Identity Management*, pp. 187-204, 2016.
http://doi.org/10.1007/978-3-319-41763-9_13
NICS Lab. Publications: https://www.nics.uma.es/publications

A Technique for Enhanced Provision of Appropriate Access to Evidence Across Service Provision Chains

Isaac Agudo[1], Ali El Kaafarani[2], David Nuñez[1(✉)], and Siani Pearson[3]

[1] NICS Lab, University of Malaga, Malaga, Spain
{isaac,dnunez}@lcc.uma.es
[2] Mathematical Institute, University of Oxford, Oxford, UK
ali.elkaafarani@maths.ox.ac.uk
[3] Security and Manageability Lab, Hewlett Packard Labs, Bristol, UK
siani.pearson@hp.com

Abstract. Transparency and verifiability are necessary aspects of accountability, but care needs to be taken that auditing is done in a privacy friendly way. There are situations where it would be useful for certain actors to be able to make restricted views within service provision chains on accountability evidence, including logs, available to other actors with specific governance roles. For example, a data subject or a Data Protection Authority (DPA) might want to authorize an accountability agent to act on their behalf, and be given access to certain logs in a way that does not compromise the privacy of other actors or the security of involved data processors. In this paper two cryptographic-based techniques that may address this issue are proposed and assessed.

Keywords: Accountability · Attribute based encryption · Auditing · Cloud computing · Proxy Re-Encryption

1 Introduction

There are a variety of data protection concerns related to cloud computing that include ongoing questions of jurisdiction and exacerbation of privacy risk through sub-processing and de-localisation, as well as legal uncertainty[1].

Auditing and verification of accounts by third parties are a necessary part of provision of accountability in complex service provision ecosystems, and the related building of trust [1]. However, the implementation of certain accountability measures themselves could introduce data protection risks, such as breach risks. For example, personal data would be included both in accounts and also in

A. El Kaafarani—Work done while at Hewlett Packard Labs.

[1] European DG of Justice (Article 29 Working Party): Opinion 05/12 on Cloud Computing (2012).

D. Aspinall et al. (Eds.): Privacy and Identity 2015, IFIP AICT 476, pp. 187–204, 2016.
DOI: 10.1007/978-3-319-41763-9_13

logs that are gathered for evidence. Hence, for both privacy and security reasons it is not the case that all actors should be able to see all logs, nor even that actors that may need to have a view on parts of certain logs in order to provide appropriate audit or verification should even be able to see all parts of those logs. This is the main problem that we address. In this paper we present and compare two novel approaches for controlling access to logs (or other forms of evidence that may be used when assessing accountability) and provide examples of their usage within cloud data transfer scenarios.

Specifically, we describe two cryptographic schemes which provide a fine-grained access control mechanism to the logged data. The different actors may be authorised to have different views on relevant log files, in various ways. For example, data subjects may disclose their data to data processors based on a privacy policy which states how their data will be processed by data processors, and who can access its related log files that are possibly distributed across multiple log servers. A data subject should be always able to access data logged about his data, but (s)he might want to delegate this right to some auditing agents of his or her choice.

2 Related Work

In general, technical security measures (such as open strong cryptography) can help prevent falsification of logs, and privacy-enhancing techniques and adequate access control should be used to protect personal information in logs. More specifically, relevant techniques for addressing accountability in the context described above include non repudiable logs, backups, distributed logging, forward integrity via use of hash chains and automated tools for log audits; for example, a log analyser and framework that link obligations right through to lower level distributed evidence in cloud supply chains that show whether or not the obligations are met [2]. Core techniques from the field of secure logging are described in [3–5]; the first paper about using chains of values of cryptographic hash functions for proofs of integrity dates from a paper about time-stamping by Haber and Stornetta [6]. Closely related research related to secure logging for accountability has been carried out within Daniel Le Metayer's group in INRIA (for example [7,8]). While secure logging techniques mainly focus on integrity (for example via the use of hash chains), our research is concerned with access control in logging systems. Of course, there has been a great deal of work on various techniques for access control, including Role-Based Access Control (RBAC) techniques, protection of data during database query [9], etc.

Our research provides contextual access to logs to governance actors (such as regulators) and other third parties. There is some analogous research carried out that instead takes a data subject centric approach, providing a transparency-enhancing tool for informing users about the actual data processing that takes place on their personal data [10]. But the range of applications is therefore smaller. Some of the assumptions and techniques vary, although there are a number of strong similarities to that approach. *Transparency Log* (TL) [11] is

a transparency enhancing tool that plays an essential role in the A4Cloud EU project[2], namely, to facilitate the communication of data from data controllers to data subjects. In order to provide *secrecy*, TL uses an encryption scheme that is indistinguishable against chosen plaintext attacks (IND-CPA) (see [12]); however, using this type of encryption scheme, an entity has to encrypt multiple copies of the same data if it wants to send them to different recipients even if they are allowed to see the same subset of log information.

Our research relies on technical means to protect evidence, whereas there is other (potentially complementary) research that aims to improve transparency (for example, by exposing additional evidence to actors that can then be assessed). An example is the Cloud Security Alliance (CSA) Cloud Trust Protocol (CTP)[3], which works at the transport interface layer and which allows cloud service consumers to ask for and receive information about "elements of transparency" from cloud service providers. Indeed, transparency is an important aspect of accountability [1]. At present, end users have an unequal relationship with service providers, exacerbated by current cloud providers offering "take it or leave it" terms of service: users have no bargaining power *vis-a-vis* cloud service providers and no means of verifying that cloud providers behave as they claim they do. Furthermore, this transparency should be provided in a way that does not impinge on privacy, especially since there can be tension between privacy and transparency [10]. The Privacy by Design approach strives to reach a positive sum, which allows both privacy and accountability/transparency to be implemented [13]. Such a positive sum can be achieved by pseudonymity schemes with the help of cryptography which allow revocation of anonymity for misbehaving users, making them accountable for their actions while guaranteeing strong anonymity for honest users (see for instance [14,15]). Previous work has been done on pseudonymisation of log files [16] for enhancing privacy of accountability tools, such as intrusion detection systems (see e.g. Revocable Privacy[4]). Further related research on privacy aspects of transparency has been conducted by the Revocable Privacy project[5], the FIDIS [17], PrimeLife [18,19] and A4Cloud EU projects.

There is a need for provenance logs for accountability, and yet there is a potential tension with privacy. This issue is not confined to cloud service provision chains, but is a broader issue that also relates to scenarios relating to internet of things, smart cities and so on, where there can be many logs reflecting what has been happening in the system. Quite often, to solve this problem, some degree of approximation could be introduced into the logs [16], but data linkage could reduce any pseudonymisation used and make it obsolete. It is not clear what the noise added in should be, and how that should match the context (and especially the purpose of usage). Deciding how to structure the information

[2] http://www.a4cloud.eu/.
[3] https://cloudsecurityalliance.org/research/initiatives/cloud-trust-protocol.
[4] http://www.cs.ru.nl/jhh/revocableprivacy/.
[5] http://www.fidis.net/fileadmin/fidis/deliverables/fidiswp7-del7.12_behaviouralbiometric_profiling_and_transparency_enhancing_tools.pdf.

and how to obfuscate it and which parts to obfuscate is non trivial and requires further research. Prior research in the area of data provenance[6] needs to be combined with new privacy preserving methodologies. The approach described in this paper could potentially be combined with such an approach: our work addresses certain tensions between log access, accountability and privacy, given existing logs within a system, rather than addressing the generation of logs in an appropriate format or the provenance of such logs.

The logs that are protected may relate to data creation, access, flow, type, deletion or handling [20]. The proposed research is not dependent on particular means used to produce logs, and could build on top of logs produced by existing Security Information and Event Management (SIEM or SIM/SEM) solutions [21]. These include, for example, RSA envision[7] and HP ArcSight[8], which provide a standardized approach to collect information and events, store and query and provide degrees of correlations, usually driven by rules. SIEM solutions do not cover business audit and strategic (security) risk assessment but instead provide inputs that need to be properly analysed and translated into a suitable format to be used by senior risk assessors and strategic policy makers. Risk assessment standards such as ISO 2700x[9], NIST[10], etc. operate at a macro level and usually do not fully leverage information coming from logging and auditing activities carried out by IT operations. Similarly, there exist a number of frameworks for auditing a company's IT controls, most notably COSO[11] and COBIT[12].

3 Scenario

We consider the aspect of controlling access to the logs (and related accountability evidence) involved within cloud ecosystems. Accountability evidence can be defined as a collection of data, metadata, routine information, and formal operations performed on data and metadata, which provide attributable and verifiable account of the fulfilment (or not) of relevant obligations; it can be used to support an argument on the validity of claims about appropriate functioning of an observable system [22]. Logging can be performed at different levels and evidence can have different sources as well [20]. In particular, at the data level we might be interested in: data creation, data access, data flow, data type, data deletion, data handling and data notification.

Since data that is collected for accountability might itself be data that can be abused, it might need to be protected as much as the processed data. The potential conflict of accountability with privacy is somewhat reduced as the focus

[6] See for example http://workshops.inf.ed.ac.uk/tapp2015/.
[7] http://www.rsa.com/experience/envision/3n1/.
[8] http://www.arcsight.com/.
[9] http://www.27000.org/.
[10] http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf.
[11] http://www.coso.org.
[12] http://www.isaca.org.

in data protection is not on the accountability of data subjects but rather of data controllers, who need to be accountable towards data subjects and trusted intermediaries, and the identities of the service providers do not need to be hidden.

As discussed in Sect. 1, we consider the subproblem of how appropriate views may be provided to selected cloud stakeholders on logs and evidence-related information. The overall scenario is shown in Fig. 1, where governance actors, or indeed the cloud subjects or the cloud customers, may need to have a view on logs or other accountability evidence produced within the cloud ecosystem. In this figure we use the extension of the NIST cloud actor roles defined in the A4Cloud project, by which in a data protection context data subjects whose information is being processed in the cloud are relevant parties called cloud subjects. In addition, there may be an organisational cloud customer with the role of a data controller, there can be more than one cloud service provider, and accountability agents may be acting on behalf of the cloud customer or of data protection authorities (a form of cloud supervisory authority). The organisational cloud customer, and indeed the cloud service providers, will have their own internal business governance. For further details, see [1].

Rules need to be enforced about who may access what aspects of the logs. We require authorized entities that may be for example a cloud customer, their clients, data subjects, auditors, cloud service providers and regulators to be given different views on the logs and evidence derived from logs, in a way that satisfies data minimization. That is to say, such that they should be able to see only the source or derived data from logs that is necessary to provide the level of assurance that they need. Furthermore, this must be without revealing personal or other types of confidential information (either in the processed data, source data logs or evidence derived from these logs) that they do not need to see.

In some scenarios it might also be needed that governance actors that oversee the way that data is handled in the Cloud (for example as shown on the left hand side of Fig. 1) delegate access to some parts of the logs in an accountable way. For example, an authorised cloud auditor might require the assistance of some other auditor. Instead of sharing the logs directly it would be preferable that access by the secondary auditor to the logs is also performed in an accountable way, subject to the same (or more constrained) rules as the primary auditor.

The specific requirements that are addressed by this scheme are as follows:

- **R1** (Secrecy). Confidentiality of the logs regarding data processing, as parts of the logs may reveal personal or confidential data.
- **R2** (Fine-grained Access Control). Data should be structured in a way that makes it possible to give different views of the logs to different entities, without necessitating an all-or-nothing approach.
- **R3** (Audit). It should be possible to track access to the different views of the logs, namely recording in a trustworthy way who accessed what and when that happened.

Confidentiality of the personal and/or confidential data processed within the cloud service provision chain is of course a very important aspect too, but is

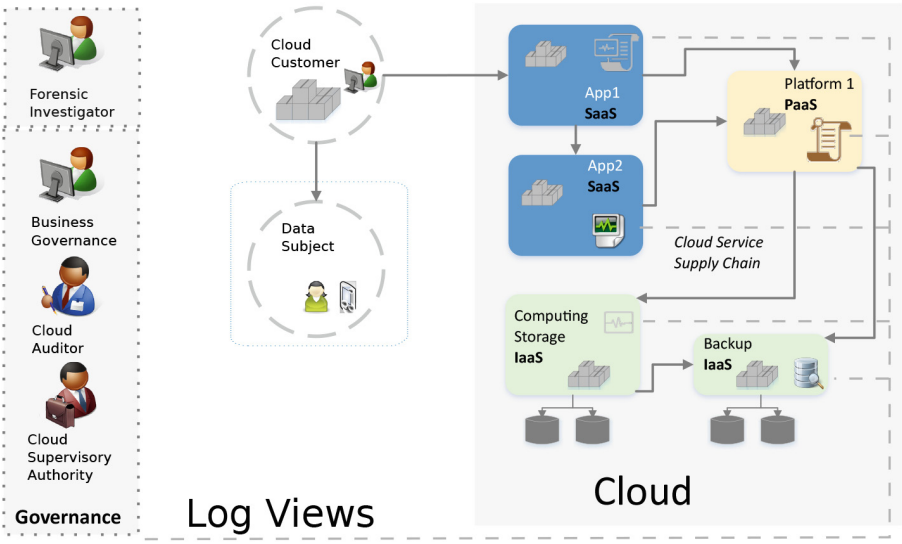

Fig. 1. High level overview of the scenario

not included in the requirements above as this problem is not the focus of our attention. Furthermore, the confidentiality of the logs regarding access to the data processing logs, i.e. auditing logs is in general much less of an issue than integrity requirements concerning those logs, so we do not here include it in **R1**.

Note that in our approach we are relaxing some of the security requirements defined in [11] as we leave some secure logging aspects (such as log integrity) to other well known techniques to address and we are not making the presumption that the logs are readable only by the associated data subject. For example, we are not covering Forward Integrity [3], i.e. the adversary should not be able to make undetectable modifications to logged data (committed prior to compromise). Another assumption we make in this paper is that cloud providers follow the honest-but-curious adversary model [23], which means they follow the established protocols, but will try to access the protected information.

We will now discuss some particular solutions that address the above requirements.

4 Solutions

We propose an alternative solution to existing transparency logs [11] that overcome some of their previously identified issues, mainly the lack of flexibility when different auditing agents with different privileges or views are involved. Also, we allow the CSP to delegate to a logging module or some other component the task of granting access to different auditing entities.

We propose a transparency-enhancing tool in the form of a cryptographic scheme that enables data processors to control access to auditors, regulators,

data subjects and other authorized parties to information about the data processing that is carried out within a service provision chain.

In this new approach, we deal with cases where data subjects might have a say in deciding *who* can play the role of an auditor, or simply when data controllers want to encrypt different levels of sensitive information presented in the logs once and yet allow more than one party to access different parts of them, based on their roles in the whole scenario. Figure 2 presents a generic diagram of our approach that shows the key components, namely, logging of the processing of personal and/or confidential data, access to those logs by authorised parties, and auditing about that data access, grouped in the following domains:

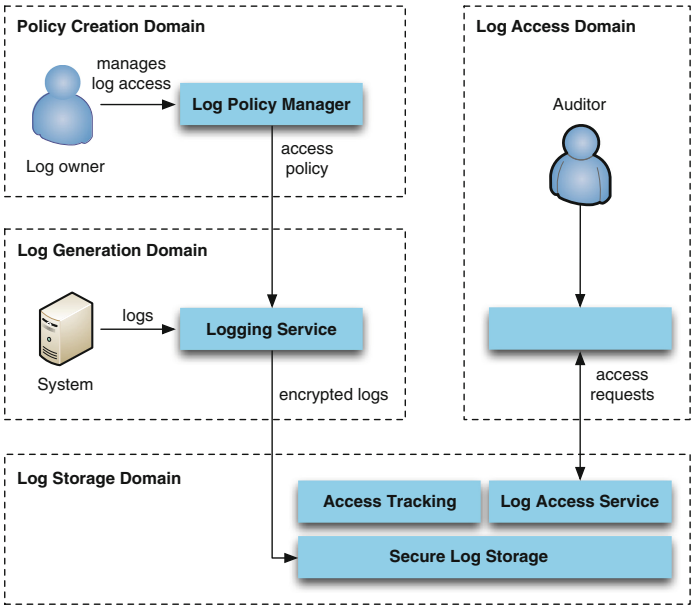

Fig. 2. Domains and processes in our solution

- **Policy Creation Domain**. This domain involves the definition of access policies to protect the logs, using a Log Policy Manager.
- **Log Generation Domain**. This domain encompasses the logged system (i.e., CSP) and an associated Logging Service, which generates the logs regarding the processing of data and protects them according to an access policy, encrypting all entries of the log or some parts of them. Depending on the kind of approach we take we might need to also categorise logs based on their contents and decide what roles would have access to them.
- **Log Storage Domain**. This domain is in charge of storing encrypted logs, and more importantly, of providing and tracking access to these logs by authorised governance actors. We are generally agnostic as to the particular cloud

storage service provided. The enforcement of access control policies is done in the Log Access Service component. In addition, for every access, there should be an audit log produced by the Access Tracking component, indicating who accessed what and when.
- **Log Access Domain**. This domain involves the access to the logs by the governance actors (e.g., auditors), using a Log Access Client.

This architecture supports different implementations. We differentiate between two main strategies for instantiating our proposal:

- ***A priori* or Agent centric**. In the a priori approach we put the focus on the agents that can audit the log. We aim at providing a solution that can *protect* log entries based on who should be entitled to access them. This way, data is protected with some embedded access control policies before reaching the data storage. When using this approach, agents can directly access the protected data and consume it according to the embedded policies.
- ***A posteriori* or Data centric**. In the a posteriori or data centric approach we put the focus on the log (or other evidence). We aim to provide a solution that can *protect* logs based on its content. This way, no policy is taken into account when protecting the log but dependent on the content, the level of protection will be different. When using this approach we need to provide an *a posteriori* mechanism for access control that will be enforced when requesting access to the log.

In the following section we introduce two implementation strategies for each of the approaches, but it would not be hard instead to implement hybrid approaches using any of the strategies. We also discuss pros and cons of both of them.

5 *A Priori* Approach: Attribute Based Encryption

A cryptographic technique which we can employ in order to provide both confidentiality and access control mechanisms is a suitable attribute-based encryption (ABE) scheme. For instance, one can use either of [24,25], knowing that in [26], they show how to construct a Verifiable Computation scheme with *public delegation* and *public verifiability* from any ABE scheme. On the other hand, Sahai and Seyalioglu provide in [27] an ABE scheme that enjoys both delegation of access rights and revocation of private keys. Similar to [11], data subjects in our approach can always have *full* access to data logged about their data. But now, they (or the data controllers) can delegate this right to some accountability agents.

All they need to do is to include them in the encryption policy part of the privacy policy. As an example of an encryption policy, one can think of the following boolean combination:

$$\Psi := (\mathsf{DataSubject}) \vee \big((\mathsf{AuditorLocation} = \mathsf{EU}) \wedge (\mathsf{BSI\ accredited})\big)$$

Encrypting logs with respect to this policy means that the corresponding data subject can access the *full* data presented in the logs, but so can any auditor who is located in EU with a British Standards Institution (BSI) accreditation. Other encryption policies will be defined for potential accountability agents, to share the access of the same subset of metadata. Some scenarios would allow data subjects and data processors to agree on the full privacy policy when they start their negotiation (if there is any). The encryption policy could be then a part of the privacy policy.

5.1 Building Blocks: ABE

Attribute-based Encryption was first presented to surpass one of the drawbacks of *sharing* encrypting data on the cloud; namely, we were able to share data *only* at a coarse-grained level; i.e. by giving away a private key to whoever we want to share our data with. Attribute based encryption offers a way for fine-grained sharing of encrypted data. Informally speaking, someone can encrypt his data with respect to a certain predicate (aka policy), and only people who have enough credentials to satisfy the predicate are granted access to the data.

In our approach we propose to use the Ciphertext-Policy variant of ABE (or CP-ABE). In this paradigm, encryption is done using a function Enc which takes a message m, the policy Ψ and any other public parameters PK established at the beginning of the protocol. The decryption key (which is issued by an attribute authority to entities satisfying policy Ψ or a suitable subset of that policy) is computed from the attributes $\mathcal{A}$ of the entity and the master key established at the beginning of the protocol. Once an entity has been issued a decryption key corresponding to its attribute set $\mathcal{A}$, it can decrypt ciphertexts associated to policy Ψ. For illustrative purposes, the following is the generic syntax of a CP-ABE scheme:

- $\mathsf{Setup}(1^\lambda, U)$: For a given security parameter, it generates a set of global parameters, represented by PK, and the set of attributes U. The global parameters include the master key MSK.
- $\mathsf{Enc}(\mathsf{PK}, \Psi, m)$: The encryption function takes as input the message m, the public key PK and an encryption policy Ψ, and produces a ciphertext CT.
- $\mathsf{KeyGen}(\mathsf{MSK}, \mathcal{A})$: Taking as input the master secret key MSK and a set of attributes $\mathcal{A}$, it creates the private key $\mathsf{SK}_\mathcal{A}$ associated to $\mathcal{A}$.
- $\mathsf{Dec}(CT, \mathsf{SK})$: The decryption function takes as input a ciphertext CT and a private key SK for a set $\mathcal{A}$. If $\mathcal{A}$ satisfies the encryption policy associated to CT, then this function returns the original message m.

The important thing to note is that decryption can only succeed in such a scheme if the entity trying to decrypt satisfies the policy that has been specified for the ciphertext, namely, if the entity holds the attributes which are required. A given key can decrypt a particular ciphertext only if there is a match between the attributes of the ciphertext and that key.

5.2 Mapping/Integration

Using ABE as building blocks, the following are the main architectural components of our approach:

- Attribute Authority (AA). There are one or more of these entities, which deal with attribute management, i.e. issuing, revoking and updating users' attributes. They create and distribute secret keys that are associated with attributes. The AAs run KeyGen, which gives a user a set of secret keys corresponding to the attributes that s/he holds, and these are distributed to the appropriate entities. This process is likely to be integrated to some extent to existing business processes, some of which may be offline. It is important that the mechanisms by which the attributes are allocated, distributed and kept up to date can be trusted by other entities in the system, but some of this process is necessarily out of scope for the proposed architecture as it will vary across different scenarios and related infrastructures.
- Trusted Authority (TA) (optional). Optionally, a central authority can be used, whereby all the users and attribute authorities in the system register with a central certification authority to obtain certificates, i.e. this entity underpins a PKI, distributing global secret and public keys. In an initial phase, the TA sets up the system by running Setup. Alternatively, a different approach can be taken by which this entity is not needed, and instead of the legal users in the system having public keys distributed by this entity, they have credentials corresponding to their attributes, issued by multiple attribute authorities.
- Log Policy Manager (LPM). This deals with policy creation and modification aspects, as well as building of lists of standardized attributes for relevant domains, which may correspond to more than one attribute authority.
- Logging Service (LS). This allows generation of a fresh symmetric key (e.g. for AES) using a random number generator, encryption of log entries using said key and encryption of the symmetric key using ABE. This hybrid encryption is used to make the process more efficient, relative to encrypting potentially large data objects directly using ABE. More formally, for a given log entry ℓ_i, LS generates a fresh symmetric key k_i and encrypts it using ABE with respect to a policy Ψ, which results in the ciphertext $c_{i,1} = \mathsf{Enc}(\mathsf{PK}, \Psi, k_i)$. The log entry is encrypted with k_i using symmetric encryption, which produces $c_{i,2} = \mathsf{SymEnc}(k_i, \ell_i)$. Therefore, the hybrid encryption produces a pair $(c_{i,1}, c_{i,2})$. Overall, LPM interacts with log owners to allow them to form/modify policies (using a standardised set of attributes for a given domain), and LS encrypts a symmetric key with this policy using ABE, that is used to encrypt the log entries and store them in the cloud storage. This component would run as a separate service, probably using the cloud storage service (CSS) to store encrypted policies and other customer related data needed for the functioning of the service (using keys generated by LS).
- Log Access Client (LAC). This process involves decryption of the symmetric key using ABE as well as decryption of log entries using the symmetric key. In response to an auditor request for access to the logs, LAC fetches the corresponding encrypted logs from CSS, and decrypts them using auditors'

secret keys, if the auditor has enough attributes to satisfy the policy Ψ. The LAC is likely to be run as a separate service, optionally integrated with LS for business reasons.

– Revocation Manager (RM). This component handles revocation and delegation. This may be handled in a number of different ways (e.g. time windows, revocation lists or more advanced techniques) and is the subject of further research which mechanisms would be most appropriate in the context of the project. Depending upon the decision about which one is chosen, aspects related to revocation may be integrated within the TA or AC, or run as a separate service.
– Cloud Storage Service (CSS). This entity stores encrypted logs, and provides them to governance actors (e.g., auditors) by means of a Log Access Services (LAS).

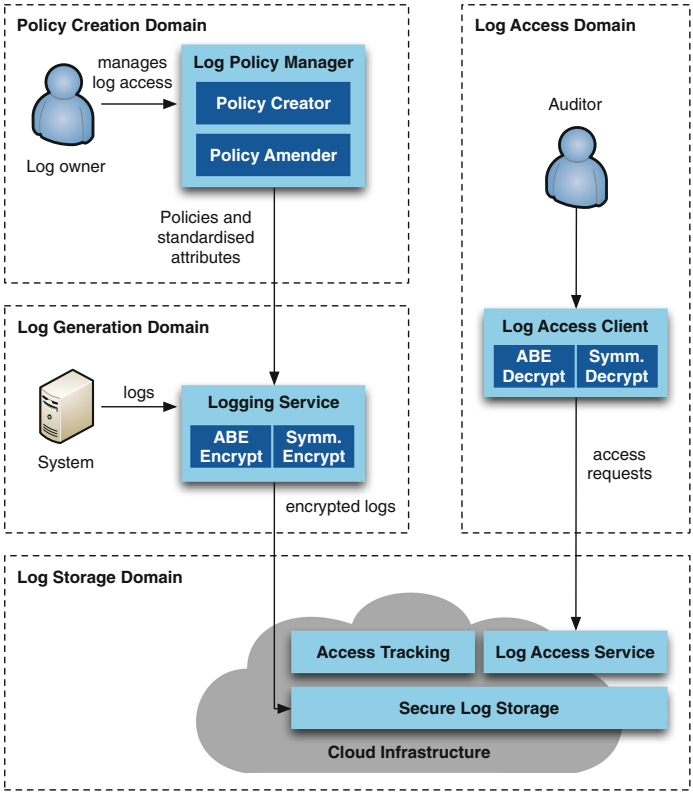

Fig. 3. ABE-based architecture

Figure 3 depicts the main components of this architecture. In addition, there can be optional integration with existing services (e.g. key storage service, authentication service).

The protocol has a setup phase and an operational phase; the purpose of the former is to establish all security parameters and to codify in machine-readable form the policy of the log owner with regards to access to its logs and encrypt a symmetric key that encrypts the log using that policy as an encryption key. The operational phase comprises the steps that are carried out in order to grant auditors access to certain logs in the cloud system in accordance with the policy that has been established in the setup phase, or subsequently modified (with associated re-encryption).

Finally, we note that we could replicate this pattern for the Tracking and Audit Access processes in case we also want to protect the audit logs.

5.3 Analysis/Assessment of Level of Fulfillment of Requirements

The requirement **R1** (Secrecy) is satisfied as long as we use an ABE scheme that achieves a proper security notion, such as IND-CPA or IND-CCA. The requirement **R2** (Fine-grained access control) is met by encrypting different parts of the log with different policies. The requirement **R3** (Audit) is fulfilled through the Access Tracking component in the Log Storage Domain, which records all access requests received by the Log Access Service.

The main advantage of this approach is that the log information does not need to be encrypted multiple times, each with the public key of the entity that is to be given access to this information, but instead just needs to be encrypted once. Also, the identity of the accessors does not need to be known in advance, so this suits scenarios where the role of an auditor that can access information (together potentially with other properties) is known, but the specific person holding this role (and satisfying any additional constraints) is not necessarily known in advance.

The main drawback of this approach relates to the necessity of a pre-phase in which attributes are distributed, and in which the actors distributing these attributes need to be trusted, as well as the attributes themselves needing to be standardised and understandable (across the actors that use them) as well as revocation mechanisms needing to be in place (although there are various options for this). In addition, if the policy relating to access changes, then the log information would need to be re-encrypted based upon that.

6 *A Posteriori* Approach: Proxy Re-Encryption

In this section we discuss an alternative approach based on Proxy Re-Encryption (PRE) [28,29], which is a type of asymmetric encryption that allows a proxy to transform ciphertexts under Alice's public key into ciphertexts decryptable by Bob; this transformation is called *re-encryption*. The main idea behind the use of PRE in our scenario is that the re-encryption process can be seen as a way of enforcing the access delegation decision, and therefore, it is an ideal place to implement the auditing capability (i.e., recording all access requests), such that it cannot be bypassed, since the re-encryption is necessary to access

the information. First we propose a preliminary version using a regular PRE scheme, and next we extend this proposal to support the requirement of fine-grained access control using a variant of PRE called Conditional PRE.

6.1 Building Blocks: PRE, Conditional PRE

As described before, PRE allows a proxy to re-encrypt ciphertexts intended for Alice into ciphertexts for Bob. In order to do this, the proxy is given a re-encryption key $rk_{A \to B}$, which makes this process possible. It can be seen that Proxy Re-Encryption materialises the functionality of delegation of decryption rights. In this case, a delegator Alice can delegate decryption rights into a delegatee Bob, and a proxy can enforce such rights by means of re-encryption. The re-encryption key act as a token of delegation. Therefore, PRE is an ideal candidate for constructing an access control system where information is outsourced (e.g., the cloud).

In our scenario, the parties that are interested on accessing the encrypted information are the consumers of the accountability evidence, in the form of logs. These consumers include auditors, data protection authorities, and data subjects, among others. Logs are encrypted under the public key of the data owner (i.e., the CSP). The CSP will grant access to specific consumers by means of re-encryption keys, which will be handed to log servers. In turn, log servers will act as proxies, enforcing the access control through the re-encryption procedure. When an auditor requests access to some log, the log server will verify if s/he is authorised (i.e., the corresponding re-encryption key exist) and will re-encrypt the data to the consumers's public key. The log servers are then capable of recording all access requests.

This solution can be extended to support a finer-grained access control using Conditional Proxy Re-Encryption (CPRE) [30,31]. CPRE is a specialization of PRE, where ciphertexts are tagged during decryption with a keyword (the "condition"); similarly, re-encryption keys are also tagged with a condition when they are generated, so only ciphertext tagged with the same condition is re-encryptable by that key. Therefore, for a given ciphertext CT_A with condition w, originally encrypted under A's public key, the proxy can only re-encrypt the ciphertext to user B if he has the re-encryption key $rk_{A \overset{w}{\to} B}$.

Using CPRE, the data owner can provide finer-grained access by means of a set of re-encryption keys with different conditions. The extension is essentially identical to the basic proposal, but ciphertexts (i.e., logs) are tagged during encryption with a specific condition (e.g., "contains personal information"). The proxies will be able to re-encrypt encrypted logs (or not) depending on whether they have the corresponding re-encryption keys. For the sake of completeness, we show here the generic syntax of a CPRE scheme:

- $\mathsf{Setup}(1^\lambda)$: For a given security parameter, it computes a set of global parameters $params$.
- $\mathsf{KeyGen}(params)$: It computes a pair of public and private keys (pk_i, sk_i).

- $\mathsf{Enc}(pk, w, m)$: For a given public key pk and a message m, it computes a ciphertext CT tagged with keyword w.
- $\mathsf{Dec}(CT, sk)$: It decrypts a ciphertext CT for a given private key sk.
- $\mathsf{ReKeyGen}(sk_i, w, pk_j)$: For a given private key sk_i, a keyword w, and a public key pk_j, it computes a conditional re-encryption key $rk_{i \xrightarrow{w} j}$.
- $\mathsf{ReEnc}(CT_i, rk_{i \xrightarrow{w} j})$: Using conditional re-encryption key $rk_{i \xrightarrow{w} j}$, it re-encrypts a ciphertext CT_i, encrypted under pk_i and keyword w, into a ciphertext CT_j decryptable by sk_j.

6.2 Mapping/Integration

In order to deploy a CPRE-based access control system for logged data, it is necessary to agree first on a set of public parameters, generated by the Setup algorithm. Next, interested parties should possess a keypair, generated by the KeyGen algorithm; these parties not only include data consumers (e.g., auditors), but also the data owner. Based on off-line authorisation decisions, the data owner can grant access to data consumers using the ReKeyGen algorithm, producing re-encryption keys that act as authorization tokens; additionally, in the Conditional PRE setting, a condition tag will be associated during this process. These re-encryption keys are handed to the log servers, which will act then as re-encryption proxies, enforcing access control. Finally, data consumers can read the encrypted data using the Dec algorithm.

Figure 4 depicts our approach based on the use of Conditional Proxy Re-Encryption. The architecture is similar to the ABE-based one, but using CPRE for the encryption and decryption of logs; consequently, the components are adapted here to this cryptosystem. A fundamental difference, however, is that the CSS now provides a Log Access Service (LAS), which implements the re-encryption functionality. It can be seen that this approach can be easily mapped to the elements shown in Fig. 2, which depicts CSP internals. In this case, the functionality of log servers, which act as re-encryption proxies, is twofold: on the one hand, they enable log access through the re-encryption procedure; on the other hand, since they control re-encryption they can record access control requests in order to support tracking and audit access. In order to make the re-encryption possible, it is necessary that the system logs generated on the CSP side are encrypted under its own key, so it can be re-encrypted to the consumer's public key.

6.3 Analysis/Assessment of Level of Fulfillment of Requirements

Requirement **R1** (Secrecy) is clearly fulfilled by using this approach. As with any public key encryption scheme, the idea is to use a hybrid encryption approach, where data is encrypted with a symmetric scheme under fresh random keys, and then these keys are encrypted with the PRE scheme. This way logged data is stored encrypted and the corresponding keys are only provided to authorised entities using public key cryptography.

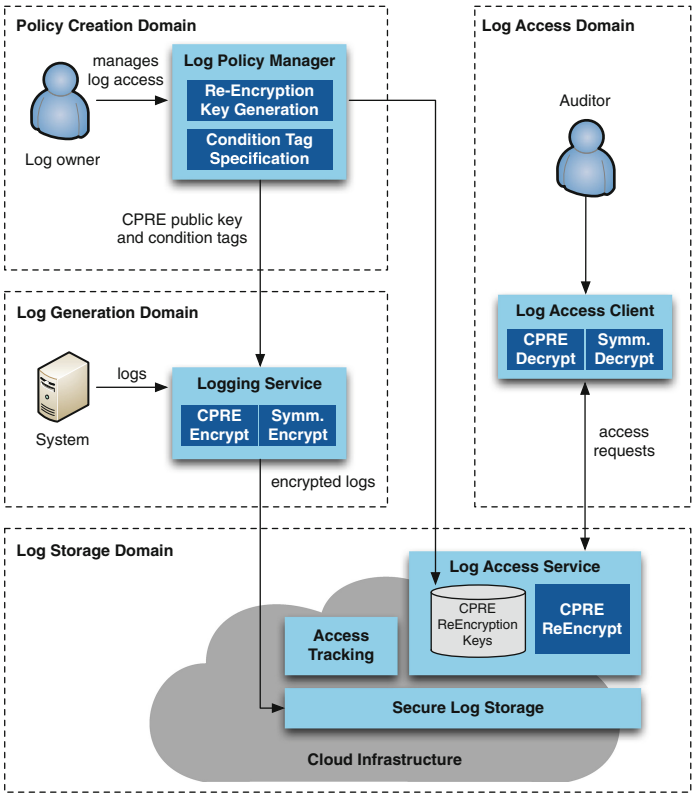

Fig. 4. CPRE-based architecture

It can be seen that this approach for access control partially supports requirement **R2** (Fine-grained access control). Using a conditional PRE scheme we can discriminate which data to grant access to (i.e., produce different logs views) and with the issuance of the corresponding re-encryption keys we can also decide who will have access to the different views of the logs.

We require that log servers perform a cryptographic operation on logs in order to ultimately grant access to other parties based on the existence of the corresponding re-encryption key. This forces us to place some extra burden on the log servers compared to the approach using ABE but on the other hand, when implemented properly, e.g., using some trusted module, it makes it easier to audit or track the actual of access to the logs because the log server is in the position of tracking access to the stored logs, by recording the re-encryption requests from auditors, supporting in this way requirement **R3** (Audit). The idea is that the same module in charge of re-encrypting will do the audit as an atomic operation. This process can even be implemented in a trusted platform module (TPM) in order to increase reliability.

7 Conclusions and Further Work

In this paper, we presented two different transparency-enhancing mechanisms that support provision of appropriate views on audit logs, based on the use of Attribute-Based Encryption (ABE) and Conditional Proxy Re-Encryption (CPRE). These approaches provide confidentiality of logs with respect to unauthorized users, and fine-grained access control over the information about the data processing that is carried out within a service provision chain (although CPRE is more limited than ABE with respect to the satisfaction of this requirement). In both approaches, different cloud actors (e.g., auditors, regulators, data subjects and other authorised parties) may be given different views of the logs.

As further work, we plan to document more fully how these schemes would work, and assess in more detail related security aspects. In addition, we are considering the potential use of yet another advanced cryptographic technique, i.e. multi-party computation, in the context of transparency logging. Briefly, secure multi-party computation can be defined as follows [32]: *n players want to compute a function of their* ***private*** *inputs in a secure way, i.e. it should guarantee both the correctness of the output and the privacy of the players' inputs, even when some players cheat.* In this case, we consider a scenario in which different accountability agents have access to different levels of sensitive information presented in a certain transparency log. Bar the data subjects, we assume that none of the accountability agents has access to the *full* log. In this case, we want these agents to be able to work *together* in order to tell whether or not there is any breach, i.e. a given service provider is compliant with the privacy policy on which he agreed with a certain data subject. This can assure that no single auditor knows *all* the sensitive information about some data subjects. This clearly is not a trivial task, and therefore it can be considered as an intriguing idea for some future work that needs more investigation on many levels; first, to fully understand the way the current logs are formed and how flexible they are in the sense that if a certain log is divided into different chunks, how would the absence of *at least* one of them help in providing further protection of the privacy of the data subjects? Second, to study the practicality of current multi-party computation (MPC) systems because MPC is a complicated cryptographic tool and extra care must be taken to give an estimate of its impact on the practicality of the whole system.

Acknowledgements. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 317550 (A4Cloud) and by the Junta de Andalucía through the project FISICCO (P11-TIC-07223). The third author has been funded by a FPI fellowship from the Junta de Andalucía through the project PISCIS (P10-TIC-06334).

References

1. Pearson, S.: Accountability in cloud service provision ecosystems. In: Bernsmed, K., Fischer-Hübner, S. (eds.) NordSec 2014. LNCS, vol. 8788, pp. 3–24. Springer, Heidelberg (2014)
2. Chaum, D., Fiat, A., Naor, M.: Untraceable electronic cash. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 319–327. Springer, Heidelberg (1990)
3. Bellare, M., Yee, B.S.: Forward integrity for secure audit logs. Technical report (1997)
4. Bellare, M., Yee, B.S.: Forward-security in private-key cryptography. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 1–18. Springer, Heidelberg (2003)
5. Schneier, B., Kelsey, J.: Cryptographic support for secure logs on untrusted machines. In: Proceedings of the 7th Conference on USENIX Security Symposium, SSYM 1998, Berkeley, CA, USA, vol. 7, p. 4. USENIX Association (1998)
6. Haber, S., Stornetta, W.: How to time-stamp a digital document. J. Cryptol. **3**(2), 99–111 (1991)
7. Métayer, D.L., Mazza, E., Potet, M.L.: Designing log architectures for legal evidence. In: 8th IEEE International Conference on Software Engineering and Formal Methods (SEFM), pp. 156–165, September 2010
8. Butin, D., Chicote, M., Metayer, D.L.: Log design for accountability. In: 2013 IEEE Security and Privacy Workshops (SPW), pp. 1–7, May 2013
9. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic databases. In: Proceedings of the 28th International Conference on Very Large Data Bases, VLDB Endowment, pp. 143–154 (2002)
10. O'Hara, K.: Transparent government, not transparent citizens: a report on privacy and transparency for the cabinet office (2011)
11. Pulls, T., Peeters, R., Wouters, K.: Distributed privacy-preserving transparency logging. In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, WPES 2013, pp. 83–94. ACM, New York (2013)
12. Pulls, T., Martucci, L.: D: D-5.2 User-centric transparency tools. In: A4Cloud (2014)
13. Camenisch, J., Groß, T., Heydt-Benjamin, T.: Accountable privacy supporting services. Identity Inf. Soc. **2**(3), 241–267 (2009)
14. Flegel, U.: Privacy-Respecting Intrusion Detection, vol. 35. Springer Science & Business Media, New York (2007)
15. Øverlier, L., Brekne, T., Årnes, A.: Non-expanding transaction specific pseudonymization for IP traffic monitoring. In: Desmedt, Y.G., Wang, H., Mu, Y., Li, Y. (eds.) CANS 2005. LNCS, vol. 3810, pp. 261–273. Springer, Heidelberg (2005)
16. Flegel, U.: Evaluating the design of an audit data pseudonymizer using basic building blocks for anonymity. In: Proceedings of SSZ, no. P-62 in Lecture Notes in Informatics, pp. 221–232. GI SIGs SIDAR and PET (2005)
17. WP7: D 7.12: Behavioural biometric profiling and transparency enhancing tools. In: FIDIS (2009)
18. WP4.2: D 4.2.2 - end user transparency tools: UI prototypes. PrimeLife (2010)
19. Hedbom, H., Pulls, T., Hjärtquist, P., Lavén, A.: Adding secure transparency logging to the PRIME core. In: Bezzi, M., Duquenoy, P., Fischer-Hübner, S., Hansen, M., Zhang, G. (eds.) IFIP AICT 320. IFIP AICT, vol. 320, pp. 299–314. Springer, Heidelberg (2010)

20. Rübsamen, T., Reich, C., Taherimonfared, A., Wlodarczyk, T., Rong, C.: Evidence for accountable cloud computing services. In: Pre-Proceedings of International Workshop on Trustworthiness, Accountability and Forensics in the Cloud (TAFC), p. 1. Citeseer (2013)
21. Nicolett, M., Kavanagh, K.M.: Critical capabilities for security information and event management technology. Gartner report (2011)
22. Agrawal, B., Molland, H., Gulzar, H., Rübsamen, T., Reich, C., Azraoui, M., Onen, M., Pulls, T., Royer, J.C.: D: C-8.1 framework of evidence. In: A4Cloud (2014)
23. Smart, N.P.: Cryptography: An Introduction, vol. 5. McGraw-Hill, New York (2003)
24. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334. IEEE (2007)
25. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
26. Parno, B., Raykova, M., Vaikuntanathan, V.: How to delegate and verify in public: verifiable computation from attribute-based encryption. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 422–439. Springer, Heidelberg (2012)
27. Sahai, A., Seyalioglu, H., Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. Cryptology ePrint Archive, Report 2012/437 (2012). http://eprint.iacr.org/
28. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. (TISSEC) **9**(1), 1–30 (2006)
29. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 185–194. ACM (2007)
30. Weng, J., Deng, R.H., Ding, X., Chu, C.K., Lai, J.: Conditional proxy re-encryption secure against chosen-ciphertext attack. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 322–332. ACM (2009)
31. Weng, J., Yang, Y., Tang, Q., Deng, R.H., Bao, F.: Efficient conditional proxy re-encryption with chosen-ciphertext security. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 151–166. Springer, Heidelberg (2009)
32. Cramer, R., Damgård, I.B., Maurer, U.M.: General secure multi-party computation from any linear secret-sharing scheme. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334. Springer, Heidelberg (2000)