IBE Applied to Privacy and Identity Management

Marco Casassa Mont, Pete Bramhall
Trusted Systems Laboratory
HP Laboratories Bristol
HPL-2003-101
May 19th , 2003*

E-mail: {marco_casassa-mont, pete_bramhall} @hp.com

IBE, identity management, privacy, confidentiality, accountability, access control, timed release

Identifier-based Encryption (IBE) is an emerging cryptography schema. As it happens for new technologies, there are ongoing debates about its usefulness and best usage. This paper describes some practical applications of IBE, in the areas of confidentiality enforcement, privacy and identity management. These applications include: a service for timed release of confidential information; a secure role-based messaging service enforcing flexible privacy constraints; mechanisms to underpin an accountable management of identity and profile information. For each of the above applications we analyse the advantages provided by IBE against traditional cryptography and we discuss open issues.

* Internal Accession Date Only
Approved for External Publication

IBE Applied to Privacy and Identity Management

Marco Casassa Mont, Pete Bramhall

Hewlett-Packard Laboratories, Trusted Systems Laboratory
BS34 8QZ, Bristol, UK
{marco_casassa-mont, pete_bramhall}@hp.com

Keywords: IBE, Identity Management, Privacy, Confidentiality, Accountability, Access Control, Timed Release

Abstract. Identifier-based Encryption (IBE) is an emerging cryptography schema. As it happens for new technologies, there are ongoing debates about its usefulness and best usage. This paper describes some practical applications of IBE, in the areas of confidentiality enforcement, privacy and identity management. These applications include: a service for timed release of confidential information; a secure role-based messaging service enforcing flexible privacy constraints; mechanisms to underpin an accountable management of identity and profile information. For each of the above applications we analyse the advantages provided by IBE against traditional cryptography and we discuss open issues.

1 Introduction

Identifier-based Encryption (IBE) [1,2,3] is an emerging cryptography schema. It provides mechanisms to deal with flexible and lightweight encryption of digital information: its underlying model require the usage of at least one trusted third party during the decryption phase. There are ongoing debates about the usefulness and practicality of IBE, especially when compared against traditional RSA-based (public key) cryptography.

This paper briefly introduces some key properties of IBE and focuses on three real-life applications of this technology, addressing problems such as the management of confidential information, privacy and identity management. The objective is to highlight areas where IBE can be successfully applied and the values that it adds.

In particular this paper describes: a service for timed release and access of confidential information; a flexible secure role-based messaging service; mechanisms to enforce privacy and an accountable management of identity and profile information.

For each application we describe some of the advantages of IBE, we compare our solution against traditional RSA-based cryptography and discuss open issues.

2 Identifier-based Encryption (IBE)

IBE is a cryptography schema [1,2,3] with three core properties:

- **1st Property**: any kind of string can be used as an IBE encryption key (public key). This "string" consists of any sequence of characters or bytes such as a piece of text, a name, an e-mail address, a picture, a list of terms and conditions, a role description, etc. Information is encrypted by using this string along with a "public detail", uniquely associated to a specific trusted third party, referred in this paper as *trust authority (TA)*. This trust authority is the only entity that can generate the correspondent IBE decryption key. It only relies on a local *secret* that is a critical resource and needs to be properly protected;

- **2nd Property**: the generation of an IBE decryption key (associated to an IBE encryption key, i.e. a string) can be postponed in time. In other words an IBE decryption key can be generated by a trust authority a long time after the correspondent IBE encryption key was created by the encryptor.
- **3rd Property**: reliance on at least a trusted third party, the trust authority, to generate decryption keys. Multiple trust authorities [16] can be used, when required, for risk mitigation.

Figure 1 shows the IBE interaction model:

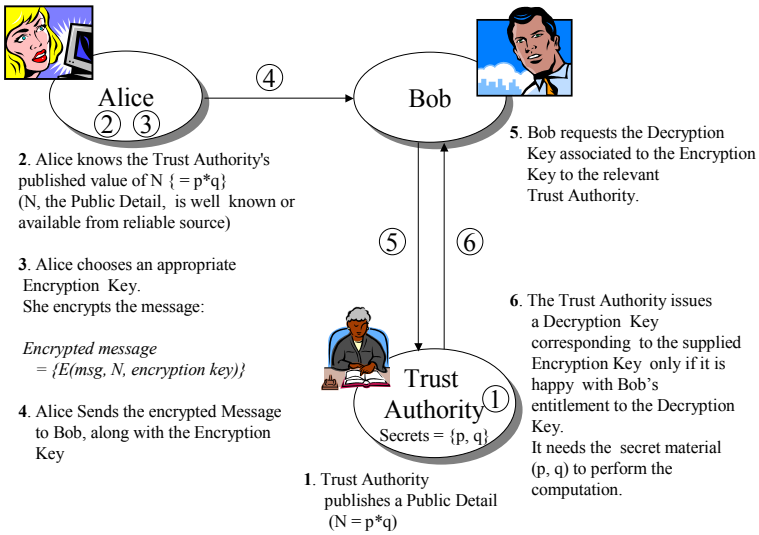

Fig. 1. IBE Interaction Diagram

Three players are involved in the above interaction model: a sender of an encrypted message (Alice), the receiver of the encrypted message (Bob) and a trust authority in charge of issuing decryption keys.

Alice wants to send an encrypted message to Bob. Alice and Bob trust a third party, the trust authority (TA). The following steps take place:

1. During the TA's initialisation phase, the TA generates a secret (stored and protected at the TA site) and a correspondent "public detail" that is publicly available.
2. Alice trusts the TA. She retrieves the public detail from the TA site.
3. Alice wants to send a message to Bob. She defines an appropriate IBE encryption key (public key) to encrypt this message. The IBE encryption key can be any type of string, for example Bob's role or Bob's e-mail address or a set of terms and conditions. Alice's message is encrypted by making use of this IBE encryption key and the TA's public detail.
4. Alice sends the encrypted message to Bob, along with the IBE encryption key she used to encrypt the message.
5. Bob needs the decryption key associated to the above IBE encryption key, to decrypt Alice's message. Bob has to interact with the trust authority to obtain this. He might have to provide additional information (credentials) to prove he is the legitimate receiver of the message.
6. The trust authority generates and issues to Bob the IBE decryption key (associated to the IBE encryption key chosen by Alice) if it is satisfied by Bob's "credentials". The trust authority might decide to generate the IBE decryption key depending on the fulfilment of specific constraints as specified by the correspondent IBE encryption key. For example a trust authority might issue an IBE decryption key to Bob only if he is compliant with a well-defined list of terms and conditions. Please notice that the IBE public key (i.e. a string), used to encrypt the document, would directly specify the list of these terms and conditions.

3 Applied IBE

This section describes three applications of IBE in real-life contexts: timed release of confidential information; a flexible secure role-based secure messaging service; accountable management of privacy and identity. Advantages against traditional public key cryptography are discussed, along with open issues.

3.1 First Application Area: Timed Release of Confidential Information

The addressed problem is the timed release of confidential information. It involves the enforcement of the confidentiality of digital documents, according to predefined time-based disclosure constraints and an efficient distribution of their content, once they become public. This problem is common in the physical world. A few examples follow:

- In the enterprise and business environment, confidential documents are generated for the consumption of managers, executive boards or working groups: often these documents can be disclosed to employees and stakeholders only at well defined points of time, dictated by trading, business and legislative constraints.
- In blind auctions a market maker can only access and disclose participants' bids at the end of the bidding time.
- In ordinary life, for example, students will know the content of their exams or their final marks only at the time dictated by the education authorities.

Prior and related work is in the area of time-lock puzzles [5,7] and timed-release cryptography [6,7].

Time-lock puzzle mechanisms are based on computational complexity. They require a precise time to solve and an intensive usage of computational resources during this time. Although this approach is interesting, it is impractical in traditional enterprise and e-commerce scenarios.

The approach based on timed-release cryptography makes use of trusted agents. Paper [6] describes a mechanism by which a confidential document is stored by the trusted agent (or by multiple agents) until its intended release time. The disadvantage of this approach is the cost in terms of resources (CPU, storage, etc.) to be used by the trust agent(s). Paper [7] proposes an alternative approach where trusted agents are not "escrow agents" as they do not have to store any information that is given to them by users. Their main task is to periodically publish a previously secret value. The disadvantage is that users must interact with a trusted agent every time they need to encrypt a confidential document.

3.1.1 IBE-based Approach

We introduce an IBE-based approach, where users do not have to interact with a "trusted agent" to encrypt confidential documents. Users perform the encryption tasks, by using their computational resources, in a stand-alone way.

This simplifies the model and introduces efficiency by reducing the number of required interactions. A "trusted agent" is still required, but its main activity is to generate and publish decryption keys: it is not affected by users' interactions.

A service can be built to provide a timed release of confidential documents by leveraging the core IBE properties. We will refer to this service as the "time vault service".

When encrypting information, IBE encryption keys (public keys) contain the disclosure date and time of the confidential document. For example, the string **"GMT200401011200"** can be used to encrypt a document and specify that its disclosure date is on January, 1st, 2004, at 12:00 noon (GMT).

The IBE decryption key is generated after the creation of the correspondent IBE encryption key. Specifically, this decryption key is generated by a trust authority (i.e. trust agent) exactly at the time of the intended disclosure of the document.

The entity that provides the above service runs a trust authority in charge of generating time-based IBE decryption keys. It has to create, store and protect the trust authority's secret. This is the only secret (independent by the number of decryption keys that are generated from it) that needs to be preserved: security efforts can be concentrated to protect it.

Figure 2 shows the high-level properties of the service:

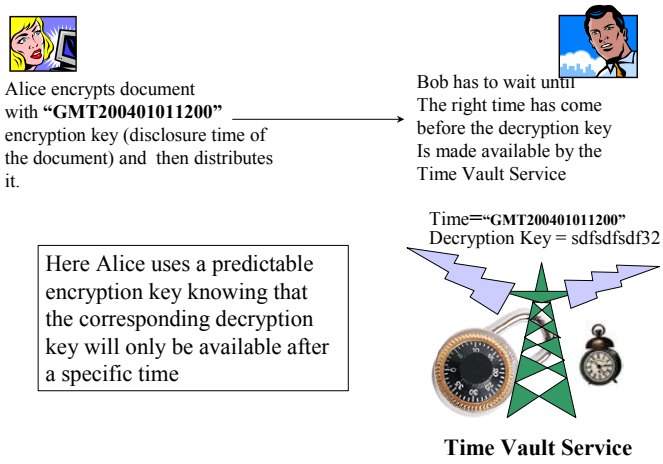

Fig. 2. Time Vault Service

The time vault service continually generates and publishes IBE decryption keys (given a predefined frequency) associated to the current time.

When Alice (the sender) generates her time-based IBE encryption keys, she must use the same (string) format adopted by the time vault service. Bob (the receiver) has to wait until the right time has come, before the IBE decryption key is made available by the time vault service.

A fully working prototype of the time vault service has been implemented as a proof of concept [8]. It consists of three main components:

- **Trusted Time Server**: it is based on an IBE trust authority. A public detail is initially created by this server and publicly made available. The correspondent secret is locally stored and secured. It *continually* issues IBE decryption keys correspondent to the current date and time. The frequency of issuances of these decryption keys depends on a predefined granularity, such as every minute or every hour or every day.
- **Distribution Service**: the distribution service is a conventional Internet/Intranet portal specialized in publishing IBE decryption keys (along with the correspondent time-based IBE encryption keys) issued by the time server.
- **Client Application**: it is the application installed at the client's site - it can be a plug-in for web browsers or e-mail browsers. It allows people to locally encrypt confidential documents and decrypt them only at, or after, their intended disclosure time. It transparently interacts with the distribution service to retrieve the IBE decryption key, if available.

3.1.2 Discussion

A similar service can be implemented by using traditional RSA-based cryptography [4,9].

Such service allows people to directly encrypt and distribute confidential documents, along with their intended disclosure time. The encryption mechanism is based on enveloping techniques, involving symmetric keys and a certified public key associated to the service (which

owns the correspondent private key). A confidential document is encrypted by a user (at their site) via a symmetric key generated on-the-fly. The symmetric key is encrypted with the public key of the service. Encrypted documents can be distributed to third parties along with metadata, which includes the encrypted symmetric key and the associated disclosure policies (an hash value derived from these policies can be encrypted along with the symmetric key, for integrity check). A receiver of the document must interact with the service to obtain the correspondent decryption key (symmetric key): the service will reveal it only at or after the intended disclosure time of the document. To achieve this it has to perform policy interpretations and cryptographic computations for every user interaction. This might introduces a computational overload.

This service will have to interpret the associated disclosure policies, on a user's interaction basis. The IBE-based service has an advantage, in terms of ***simplicity*** and ***efficiency***.

A time server based on **traditional cryptography**, has to perform interpretation and cryptographic operations for each user interaction, before releasing a decryption key. The "distribution service", associated to the (RSA-based) time server, must interpret documents' metadata to verify if the constraint on the associated disclosure time is satisfied. In such a case, the time server must perform a decryption to retrieve the symmetric key used for encryption. These two components need to interact and exchange information. This might cause delays especially during peaks of users' requests and introduce security vulnerabilities. Caching mechanisms and replication can be put in place to mitigate part of the problem but this increases the complexity of the overall service.

The time server based on the **IBE cryptography** is simpler to run as it is modular, each component is self-contained and interactions about these components are reduced to the minimum. The time server activity is completely independent of users' requests for IBE decryption keys. It is predictable. Its only allowed interaction is with the distribution service by means of an outgoing connection. Because of these aspects it is easier to protect.

It is also potentially more efficient. In our IBE-based model, users' interactions with the distribution service do not trigger any interpretation of "time-based disclosure policies" and cryptography computations, whilst this happens for the RSA-based model. Even in case of peaks of requests, the IBE-based distribution service only need to execute simple database queries and return a few hundred bytes to the client (the size of the IBE decryption key). If compared to the trusted agent described in [7] our method has the advantage of not requiring users to interact with the time server (i.e. trusted agent) at the encryption time (although this is true also for the RSA based approach).

The potential overhead of the IBE-based service is due to the fact that the time server has to continuously generate IBE decryption keys, even if they are not required. Nevertheless, this activity exclusively depends on a predefined issuance rate of decryption keys: it can be addressed during the configuration phase of the time server.

3.2 Second Application Area: Flexible Role-based Secure Messaging Service

The addressed problem is the enforcement of the confidentiality of private information in dynamic contexts, where people's roles and permissions are subject to frequent changes.

In general, (explicit or implicit) policies dictate the terms and conditions under which confidential information can be accessed or disclosed. An important aspect of the problem is making sure that these policies are enforced and cannot be subverted.

We analysed a real-life health care environment. User populations in health care organizations are usually very dynamic. General practitioners, other doctors and specialist consultants are considered precious resources: they are allocated, on demand, to specific patients' health problems. This dynamism has strong implications on how patients' confidential data has to be managed and how privacy is preserved. We partnered with a UK health care provider.

We focused on confidential information exchanged via electronic messaging services, such as the e-mail service, because of its relevance on modern organisations.

Figure 3 shows a high-level interaction model involving general practitioners (GPs) and members of a department of the health care organization:

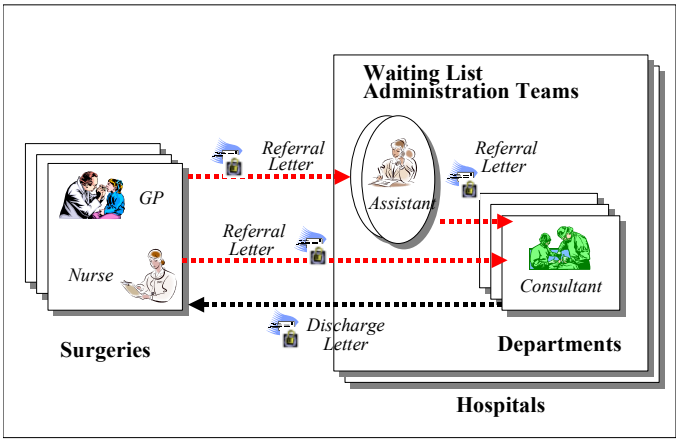

Fig. 3. Interaction Diagram

A general practitioner (GP) or his/her assistants might need to send referral letters to hospital consultants containing patients' confidential information. The GP might have no idea of which specific consultant is on duty or which specific person needs to be contacted. On the other hand, the GP has clear in his mind the role of the person he/she is willing to communicate to. Waiting list administration teams (at the hospital premises) are in charge of dealing with health care requests for patients, and allocate them to the available resources. Members of the waiting list administration teams could be asked to act as information "routers". In particular circumstances, they might not be allowed to access or read the content of the requests because of their confidentiality.

Employees' roles can be dynamically (re)-allocated, depending on timetables, availability or lack of medical personnel. There is a need for a system that protects the content of confidential messages by means of flexible disclosure policies.

Traditional cryptography [4] based on public/private keys, symmetric keys and X.509 Public Key Infrastructure (PKI) [9] can be used to address the problem. Digital certificates along with PKI infrastructures are a suitable technology to underpin authentication, non-repudiation and confidentiality. On the other hand, it is known that PKI suffers of many problems, including lack of flexibility and scalability. Certificate lifecycle management can be very "expensive" especially in dynamic contexts.

In case of secure messaging services, digital certificates and traditional (RSA-based) cryptography offer a viable solution when privacy criteria depend on the "identity" of the message receivers: in this case a confidential message can be encrypted by directly using the public key of the receiver. S/MIME is based on this principle.

If privacy criteria *do not* depend on receivers' identities but on other aspects, such as the satisfaction of predefined disclosure policies - including membership of roles - it is not possible to use digital certificates as specified before. In this case it is not necessarily known, *a priori*, who the receiver is, i.e. which digital certificate (public key) must be used for encryption purposes.

To solve the problem, a further level of indirection is required. An additional system component can to be introduced to deal with policy interpretation and authorization. This component must be a trusted element of the system. A digital certificate (well known by the system users) can be associated to this trusted component. It can be used to encrypt confidential information along with its disclosure policies. Encrypted message bundles can be created to represent, transmit and store secured messages.

To build the above solution, traditional secure e-mail services, based on S/MIME and users' digital certificates, are of little help. Additional mechanisms need to be implemented to

deal with the authoring of disclosure policies, the management of encrypted bundles and late binding of roles. A trusted component has to be built from scratch.

Although it is possible to address the problem by building hybrid solutions based on traditional cryptography and PKI (coupled with RBAC) this is not the most natural way of making use of these technologies.

3.2.1 IBE-based Approach

We introduce an IBE-based approach where confidential e-mails are directly encrypted by means of textual strings, representing IBE encryption keys. These strings explicitly describe the disclosure policies (terms and conditions) under which the content of an e-mail can be disclosed: specifically these policies contain list of roles. For example if a GP wants to send a confidential e-mail to any person that is a consultant, he/she can simply use the *"Consultant"* string to encrypt the e-mail. If a GP wants to send an e-mail that can be accessed by any member of the waiting list administration team, he/she can use the *"Member of the Waiting List Administration team"* string to encrypt the e-mail.

We make use of a trust authority (trusted third party). The receiver of a confidential e-mail has to authenticate and interact with this trust authority to retrieve the appropriate decryption key. The trust authority retrieves up-to-date lists of roles associated to users and checks them against the relevant disclosure policies. As for traditional RBAC system, in this model it is necessary to manage the associations of people's identities with their current roles.

The trust authority will generate and issue a decryption key only if the requestor has the required role(s) at the time the decryption key is requested. If the disclosure policies are tampered with, the generation of the correct decryption key is impossible.

We developed a fully working solution for the trial with the UK health care organization [10]. Its core architectural components include:

- ***An e-mail browser Add-In***: it is a standard e-mail browser add-in containing a module which provides the IBE cryptographic algorithms to encrypt and decrypt e-mails; a graphical UI to help users to easily author role-based encryption policies; a secure communication module to remotely interact, via https protocol, with the trust authority, in order to retrieve lists of roles and ask for IBE decryption keys.
- ***A Trust Authority Service***: it is a web service, providing "trust authority" functionalities (such as generation of IBE decryption keys). It is hosted by a secure and protected resource (server).
- ***A protected and secured database***, associated to the trust authority. It containing up-to-date lists of roles and up-to-date associations of people's identities to their current roles. Trusted administrators run and update the database.

The add-in uses IBE cryptography to encrypt confidential messages. All the elements of a confidential e-mail are encrypted, including its subject line, body and attachments. The add-in automatically associates the authored disclosure policy to the current e-mail. Our tool can be easily extended to allow the authoring of more complex policies.

At the receiver side, the add-in manages all the interactions with the trust authority service, to retrieve the required IBE decryption key. The trust authority shares the semantics of the disclosure policies with the add-in. After authenticating a user, it checks if the user has the required role(s) by looking at tables in an associated database. Only in case of success, it generates (on-the-fly) the decryption key. *The trust authority never accesses the content of confidential messages, as only disclosure policies (i.e. IBE encryption keys) are sent to it.*

3.2.2 Discussion

We believe that the proposed solution has advantages in terms of *simplicity* and *flexibility* against a similar solution build by using traditional cryptography and PKI infrastructure.

In terms of *flexibility*, our solution allows the encryption of confidential e-mails without knowing, *a priori*, the receiver's identity. It uses disclosure policies directly as IBE encryption keys. These keys are self-explanatory and describe "the constraints" to be satisfied by secured e-mail readers. Because of the IBE properties, it is straightforward to implement a mechanism that supports "late-bindings" of roles.

Our solution is *simple to manage*. No complex enveloping techniques need to be used. No public key/digital certificates need to be issued, managed and revoked, *at least* for encryption purposes. In the current solution, no secret needs to be stored at users' PCs or exchanged among them. The add-in (installed at the users' sites) only needs to know what the trust authority's public detail is: this is necessary for encryption purposes. It can be locally stored (at the add-in installation time) or downloaded from the trust authority web site.

The solution deployed in the trial relies on Microsoft Windows authentication mechanisms and its trust domains to authenticate users, because of constraints imposed by the hosting health organisation. This is a specific approach to authentication and it simplified the way we solved the problem.

At the current stage of our research we are exploring IBE-based challenge/response schemas for authentication purposes but we do not yet have evidence that they are better than traditional PKI-based authentication or can simplify users' experiences.

In general, IBE is a complementary technology to traditional RSA-based cryptography and it can be used to address privacy management issues mainly by exploiting its encryption features. The trial will provide us with valuable evidence about the validity and scalability of our solution.

3.3 Third Application Area: Accountable Management of Identity Information

We addressed the problem of making organizations more accountable when dealing with people's personal information and giving more control to people over their information.

We refer to an e-commerce scenario. In no way are the aspects we highlight limited to this sector, as they are common to financial, government and enterprise areas. In this scenario users deal with electronic transactions that span across multiple e-commerce sites. A person initially provides their digital identity and profile information to an e-commerce site to access their web services, possibly after accepting (or negotiating) a set of privacy and data protection policies. It might happen that other web sites or organizations need to be involved (such as suppliers, information providers, government and financial institutions, etc.) to fulfill the specific transaction. The involved e-commerce sites might have no prior agreements with the user or belong to the same web of trust.

The user might be conscious of this or this might happen behind the scenes. Little has been done so far to directly involve users, or entities acting on their behalf, in the management of their privacy, especially when multiparty interactions and transactions take place.

Users lack control over their personal information after initial disclosures of personal information. Other involved parties (such as delegates, e-commerce sites or enterprises) also lack control over the confidential information they manage on behalf of their customers, in particular when they disclose it to other organisations. It is hard to make organizations accountable for their behaviours.

A great deal of work has been done in this area to provide a legislative framework. However, privacy and data protection laws are complex and hard to enforce, especially when personal information spread across national boundaries. In general, users have little understanding or knowledge of these laws and their implications.

There is shortage of tools and mechanisms that allow users - or trusted third parties acting on their behalf - to explicitly define their own privacy policies and control their fulfillment.

Mechanisms such as W3C's Platform for Privacy Preferences (P3P) [11] allow users to define simple privacy policies but only for point-to-point interactions. There is little control over the subsequent fulfillment of these policies.

Liberty Alliance [12] and Microsoft [13] efforts in federated identity management are (for the time being) based on a closed web of trusts. Identity providers must be part of trusted clubs and be compliant with predefined privacy policies. This approach limits scalability and flexibility of the allowed interactions and transactions.

Relevant work towards a more fine-grained control over the "privacy management" of personal information has been described by [17, 18]. In paper [17] the authors define a privacy control language that includes user consent, obligations and distributed administration to describe privacy policies i.e. "sticky policies". In paper [18] the authors describe a platform for enterprise privacy practices (E-P3P). When submitting data to an enterprise, a user consents to the applicable privacy policies – sticky policies - along with selected opt-in and opt-out choices. Sticky policies are strictly associated to users' data and drive access control decisions and privacy enforcement at the enterprise site.

Papers [17,18] do not describe how the association of sticky policies to confidential data is enforced, especially when this data is exchanged across enterprise boundaries. Users need to trust the enterprise when disclosing their data. Leakages of personal and confidential information might still happen, despite data protection laws and privacy policies, because of lack of security or the dishonesty of some of the involved intermediaries.

3.3.1 IBE-based Approach

We introduce a privacy model to address the accountability and user control issues, by extending [17,18]. IBE technology plays a key role in the implementation of this model. The proposed model has the following properties:

- **Enforcement of confidentiality**: obfuscation of confidential (personal) information before its disclosure, to protect its content;
- **Strong association of privacy policies to confidential data**: association of "tamper resistant" sticky policies to obfuscated data, defined by users or trusted third parties acting on their behalf. Any tampering with these policies will prevent the access to the content of obfuscated data;
- **Policy compliance check by trusted third parties**: any disclosure of confidential data is subordinated to the fulfillment of sticky policies' constraints. This is checked and enforced by trusted third parties (Tracing and Auditing Authorities);
- **Accountability management**: auditing and tracing of disclosures of confidential data via trusted third parties (Tracing and Auditing Authorities);
- **User involvement**: active involvement of users (if desired) during the process of disclosing of their confidential data.

Figure 4 shows the architecture of a system based on it. Messaging protocols (1)-(4) are carried out in order, and involve transfer of the information indicated in the directions shown by the arrows.

Identity or profile information is encrypted with "sticky policies" (1), before its disclosure to third parties, by web browser plug-ins or trusted applications.

These policies are used as IBE encryption keys [14] and might include: references to logical names of identity and profile attribute(s); disclosure constraints; obligations; actions (i.e. notification of the owner in case of multiparty disclosure); expiration dates, etc.

To obtain a valid IBE decryption key (2), the receiver needs to interact with trust authorities (TAs) and provide information (including authentication credentials, business related information, company/individual policy related to data disclosure, usage and storage, software state, platform configuration etc.) as required by the disclosure policies. In doing this, the receiver is explicitly aware of (and understands) these policies.

A TA will issue an IBE decryption key (4) if it acknowledges the compliance with the disclosure (sticky) policies. Before doing this it might interact with the information owner (3) to ask for his/her authorization or to send a notification. The TA traces and stores all the information exchanged during these interactions in audit-trails, as evidence for future contentions or forensic analysis.

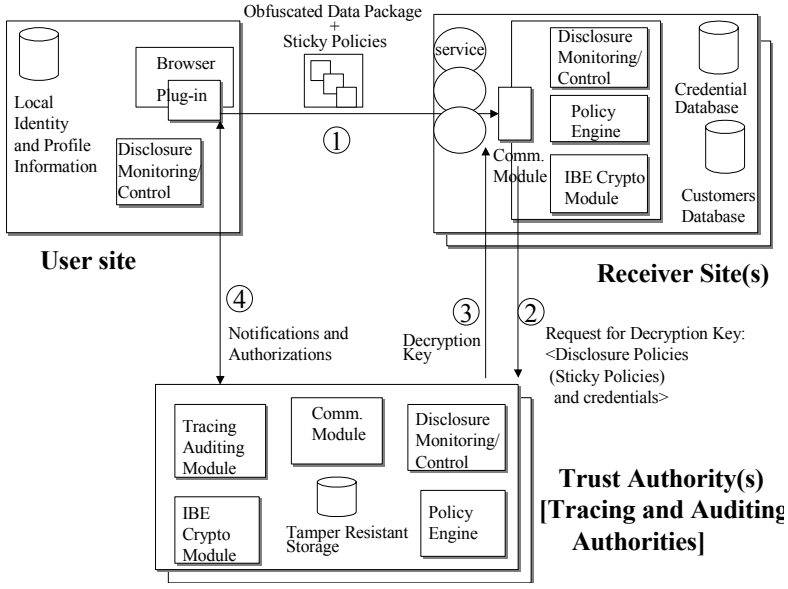

Fig. 4. High Level Architecture

Users' identity and profile information is exchanged by means of data packages containing encrypted data and their associated sticky policies. A simple example of such data package, along with a sticky policy, is described in [14].

Sticky policies can specify two categories of constraints:

- **"soft" constraints**: they are enforced via trust authorities (TAs). They can potentially be violated, once users' information has been disclosed. The involved risks are mitigated (and accountability underpinned) by TAs' tracing and auditing mechanisms.
- **"strong" constraints**: they are strongly enforced by trusted platform mechanisms such as the TCPA integrity checking mechanisms [15].

The importance of user's data dictates which "mixture" of the two categories of constraints needs to be used.

Engines and access control mechanisms are usually used to interpret and enforce privacy policies, once data has been disclosed. In our model, part of the enforcement can be done upfront, before any personal data is disclosed, if specified by sticky policies. For example, this activity could consist of checking if the IT platforms that will handle personal data satisfy basic security requirements, such us platform components' integrity.

TCPA integrity checking mechanisms [15] can be used to allow the TA platform to be checked out by the user (to make sure that the TA will operate as expected) and/or the recipient of the data (to help the recipient decide whether the TA can be trusted with the information that the recipient needs to provide to the TA in order for the decryption key to be issued).

Receivers of personal and confidential information need to be compliant with sticky policies and their constraints, as defined by the information owners. They need to make the required steps in order to demonstrate their compliance to the relevant TAs.

If the receiver of confidential data discloses it in a way that is not contemplated by the policies they previously agreed, there is an audit trail (at the TA(s) site(s)) showing that they actually understood and agreed with those policies.

In case of identity or profile thefts, the audit information can be used to pin down a list of potential "offenders" and carry on forensic analysis. Enforcing the tracing and auditing of disclosures makes the information receivers more accountable.

To enable an electronic transaction involving user's confidential data, the receiver might send obfuscated data or any portion of it to another third party, for example a service provider. It might decide to encrypt portions of this data by using additional policies. This third party has to interact again with a TA as described above. This receiver may have to use multi-

ple TAs in order to access the data. For example, one TA might be competent with respect to security platforms and other might be competent in privacy, so it would make sense for both to carry out checks before allowing an entity to access data. In this case, the user might encrypt the data using a disclosure policy that specifies that it is necessary to use two (or more) IBE partial decryption keys in order to decrypt the data, and each of the TAs would provide one of these keys. Multiple keys might be needed to decrypt the same piece of data, or different data fields might be encrypted using different keys.

The IBE encryption schema supports the management of encryption keys where the correspondent decryption keys are the result of the composition of partial decryption keys obtained by multiple TAs.

Owners of identity and profile information can run their own TA services to have first hand understanding of what happens to their information and make ultimate decisions. Alternatively users can periodically interact with the TA to monitor the disclosure state of their confidential information.

3.3.2 Discussion

We believe that the value IBE brings in this area is in the mechanisms to associate "tamper resistant" disclosure policies ("sticky policies") to confidential data and the active interaction model to force requestors to be traced (audited). Although this can be achieved also by using traditional RSA-based cryptography and PKI, IBE simplifies the way sticky policies are associated to personal data and directly used for encryption purposes. In addition the trust authority role fits well in the overall model.

The idea of using trusted third parties to mediate the access to confidential information is not new. There are well-known related issues, including why a person or an organisation should trust a third party. In our specific case, multiple trust authorities can be used to mitigate risks. In this case, the approach based on IBE is straightforward [16] because of the easiness by which multiple partial decryption keys can be combined in the IBE decryption key.

The trust authority is the right place to manage accountability, via tracing and auditing functionalities. Requestors do need to interact with the trust authority to obtain an IBE decryption key. They need to provide their contextual credentials, as mandated by the disclosure policies (sticky policies): this information is logged and can be used to make them accountable. The auditing and tracing effort is effective also to audit users' behaviors, as the trust authority is a trusted bridge between users and enterprises.

The usage of cryptography and, specifically, encryption mechanisms to preserve the confidentiality of personal data is not new. IBE technology simplifies the management of obfuscated data at the client side (no certificates need to be used and managed) and pushes complexity to the enterprise side. This removes a major practical burden from clients' perspectives.

4 Final Remarks

The IBE technology is useful and practical when flexible encryption and reliance on trusted third parties (i.e. trust authorities) are required. This is a common pattern for the three areas we explored but it might not be the case for other areas. IBE suitability needs to be evaluated case by case.

In the previous sections we discussed how IBE can add value in three areas and compared it against traditional cryptography. We can summarise our comments by saying that IBE simplifies end users' experiences by avoiding them having to deal with digital certificates (for encryption purposes) - if possible - and providing them with great flexibility in terms of encryption criteria.

In the three areas we explored, we proposed IBE-based solutions: similar results can be achieved by using traditional cryptography but at the price of higher complexity.

The usage of IBE technology can simplify the provision of services, at least from users' perspective and move the complexity to the service provider's side. This aspect is particular important in cases where critical tasks need to be achieved in simple ways, such as confidentiality management, privacy and identity management.

The IBE model relies on the usage of at least one trust authority. On one hand this introduces accountability to the system: on the other hand it could cause problems, when trust needs to be established or when trust authority's "key escrow" capability is unacceptable. The usage of multiple trust authorities can mitigate the involved risks.

In general IBE fits very well in areas where trusted third parties have a well-defined role. Its usage becomes harder to justify where no trusted third party is required or where their role is not so clear or acceptable.

At moment it is not yet clear if IBE can provide any added value in terms of authentication, especially when non-repudiation is required. Users would have to store and manage an associated IBE secret, exactly as it happens with traditional RSA-based cryptography (and PKI). The presence of a trusted third party could be a problem because of its "key escrow" capability and the implications in terms of non-repudiation.

IBE is a complementary technology to traditional public key cryptography: it has its best usage in dynamic contexts, with widespread populations, such as the end-users of services. On the other hand, public key cryptography's best usage is for more static contexts with well-established relationships between the involved parties, including service providers and trusted third parties. IBE and public key cryptography can be used together, to build hybrid solutions, in which, for example, a traditional public key cryptography can be used to support a population of trust authorities.

5 Conclusions

Identifier-based encryption (IBE) is an emerging cryptography schema. Its potential capabilities are still to be fully explored. This paper describes three practical exploitations of IBE in areas involving confidentiality enforcement, privacy and identity management.

IBE provides clear advantages to end-users, in terms of flexibility and simplicity of encryption, when one or more trusted third parties are required. At the moment it is not yet clear if IBE can add value in terms of users' authentication, especially when non-repudiation is a must. In general IBE should be considered as a complementary technology to traditional public key (RSA-based) cryptography.

References

1. D. Boneh, M. Franklin, Identity-based Encryption from the Weil Pairing. Crypto 2001, 2001
2. C. Cocks, An Identity Based Encryption Scheme based on Quadratic Residues. Communications - Electronics Security Group (CESG), UK. http://www.cesg.gov.uk/technology/id-pkc/media/ciren.pdf, 2001
3. L. Chen, K. Harrison, A. Moss, D. Soldera, N.P. Smart, Certification of Public Keys within an Identity Based System, Proc. 5th Int. Information Security Conference (ISC), 2002. LNCS 2433, Springer-Verlag, 2002
4. W. Diffie, M.E. Hellman, New Directions in Cryptography, 1976
5. J. Garay, M. Jakobsson, Timed Release of Standard Digital Signatures. Financial Crypto, 2002
6. T.C. May, Timed-release crypto, February 1993
7. R.L. Rivest, A. Shamir, D. A. Wagner. Time-lock puzzles and timed-release Crypto. MIT laboratory for Computer Science. MIT/LCS/TR-684, 1996

8. M. Casassa Mont, K. Harrison, M. Sadler, The HP Time Vault Service: Innovating the way confidential information is disclosed, at the right time, HPL-2002-243, 2002
9. R. Housley, W. Ford, W. Polk, D. Solo, RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL profile, IETF, 1999
10. M. Casassa Mont, P. Bramhall, C. R. Dalton, K. Harrison, A Flexible Role-based Secure Messaging Service: Exploiting IBE Technology in a Health Care trial. HPL-2003-21, 2003
11. W3C, The Platform for Privacy Preferences 1.0 specification (P3P 1.0). http://www.w3.org/tr/p3p - W3C Proposed Recommendation, 2002
12. Liberty Alliance Project, http://www.projectliberty.org/, 2002
13. Microsoft, Microsoft .NET Passport, http://www.microsoft.com/netservices/passport/ , 2002
14. M. Casassa Mont, S. Pearson, P. Bramhall, Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services, HPL-2003-49, 2003
15. S. Pearson (ed.), Trusted Computing Platforms, Prentice Hall, 2002.
16. L. Chen, K. Harrison, D. Soldera, N.P. Smart, Applications of Multiple Trust Authorities in pairing Based Cryptosystems. LNCS 2437, pp. 260-275, Infrastructure Security 2002, 2002
17. G. Karjoth, M. Hunter, A Privacy Policy Model for Enterprises, IBM Research, Zurich - 15th IEEE Computer Foundations Workshop, June 2002
18. G. Karjoth, M. Schunter, M. Waidner, Platform for Enterprise Privacy Practices: Privacy-enabled Management of Customer Data - 2nd Workshop on Privacy Enhancing Technologies, Lecture Notes in Computer Science, Springer Verlang – 2002