International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Managing Identities from the Perspective of Entities

Shivani[1], *Dr. Rupali Ahuja*[2]

[1]MTech Student
[2]HOD CSE Department

ABSTRACT

Entities (such as users or services) must verify their identities to have access to the services provided by service providers (SPs). A company's identity is revealed to an SP in the form of contact details.

In the traditional IDM paradigm, each programme keeps its own list of users' IDs. Multiple accounts belonging to the same company may be hosted by the same service provider (SP). If the same entity's personally identifiable information and associated attributes are used across services, it may be feasible to "map" the information back to the entity.

We propose an entity-centric approach to IDM in the cloud. The method is built on two pillars: (1) anonymous identification, which mediates interactions between the entity and cloud services using the entity's privacy policies, and (2) active bundles, which contain a payload of personally identifiable information (PII), privacy policies, and a virtual machine that enforces the policies and uses a set of protection mechanisms to protect itself.

The method's key benefits are its independence from other parties, its limited data sharing with the SP, and its support for the usage of identity data on untrusted sites.

Keywords- active bundles; cloud computing; identity management (IDM); personally identifiable information (PII); anonymous identification; zero-knowledge proofs (ZKP); privacy-enhancing technologies (PET); privacy; security.

Introduction

Managing Individual Identities

One's identity is composed of their own characteristics. Labelling something with a name or number is referred to as identification [1].

To complete the authentication process using SPs, a unique identifier for the entity is needed. A service provider (SP) can use an entity's identification to verify that the requesting entity is who it claims to be.

It's possible for a single entity to have several identities in cyberspace. Using an IDM, all of your different online identities may be managed with ease. The best way to divulge personal information in order to get a service is likewise determined by this metric. IDM is used for the following [2] purposes:

1) Establish identities by associating IDs with preexisting things.

2) A thing's defining features are the attributes you give it.

3) Monitor your usage of sensitive data: System user ID activities can be recorded, or logs can be made available.

4) Schedule the permanent erasure of all individual data. No one will be able to access the information after the deletion date.

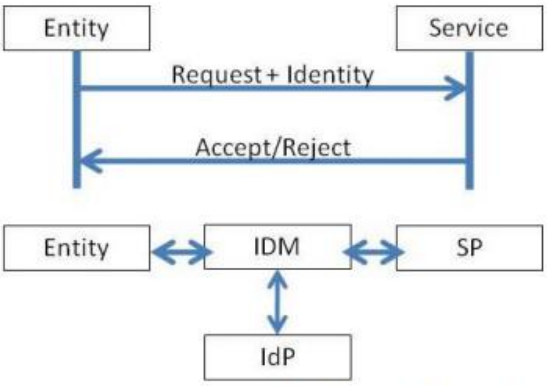

Figure 1. Authentication using Third Party Identity Management

In the authentication situation depicted in Figure, personally identifiable information (PI) was employed. A user wants to use a service but is reluctant to do so since she must provide personal information to the SP. The SP needs PII in order to correctly identify the user. Choosing what, if anything, to disclose and how to communicate it is the biggest issue.

Collaborative groups employ IDM to figure out who or what something is. Included in this set (according to [3]) are:

The primary Identity Provider (IdP) service.

It is employed in the formation of virtual personae. Companies like credit card issuers and governments alike provide individuals with identities that may be used to make transactions.

The SP (service provider).

It provides assistance to people who have the required identification. For online tax filing via SP, for example, the user is required to provide identifying information.

Real thing.

People or objects that are assumed to exist. Anything from a name and birthdate to a Social Security number might be used as proof of identity.

Identifier that ensures uniqueness.

SP may ask it to verify a claim made about a specific party. It checks the validity of the assertion to see if it is true.

A typical IDM will utilise one of the three identifiers below:

Information known by the entity and the SP, information known by the entity and the SP and verified by the IdP, and information about the entity, such as fingerprints, can all be used for authentication.

When discussing cloud computing, the term "privacy" is used to describe the user's or business' ability to decide whose data is kept in the cloud and who has access to that data. There are a variety of privacy rules in place that cloud service providers must follow when it comes to the collection, storage, use, and disclosure of user data. The security of private information in the cloud is an issue at every social level. The safety of your data and the identities of those with access cannot be determined by your cloud service provider [4].

Organisations other than the one being serviced, known as cloud service providers (SPs), store data for it in the cloud. Concerns concerning privacy and security can arise wherever data is stored in the cloud, whether by an individual, a business, a government agency, or another entity [4]. Trusting an outsider means gambling that they will act in your best interests, which isn't always the case. Here are just a handful of the many serious risks that come with using cloud services: It's possible that (i) the projected damages from a single breach may be rather high, and (ii) the diversity of "users" offers the possibility of several simultaneous threats. Some of the major problems with this paradigm are:

1) Relying on SP to get to one's own information, software, and hardware. User access control rules and security policy enforcement are also managed by IaaS. The user must have trust in the provider's ability to secure their data, maintain their internet access, and monitor activity.

2) Reluctance to place faith in others since doing so would require taking a risk. Trust and risk are, in essence, diametrically opposed ideas. Some sort of monitoring or auditing tools would be required to increase trust.

3) They are all sharing the same amenities, renters in a multi-tenant property sometimes have competing interests. Tenants have a right to some personal space.

The Cloud's Role in Managing User IDs and Passwords

The standard, app-centric [5] framework treats each application separately. The IDM architecture keeps track of its users in its own database. In cloud computing, a company may have many accounts with different SPs. In addition, businesses can use many services from the same SP (Google, for example, provides both Gmail and Google Docs). A user must give personally identifiable information (such as name, email address, and phone number) while requesting cloud services. This generates a digital footprint that, if not protected, might be exploited for malevolent reasons, such as identifying or tracking down a specific individual. It may be feasible to "map" PII to an entity if the same object's PII and related properties are used across services [5]. The biggest worry is that thieves may get unauthorised access to people's personally identifiable information (PII) [4].

In cloud computing, it is the data's owner's obligation to ensure the privacy of their data. The owner can't do this challenging task without the assistance of technological controls.

Requests for cloud services from any given business must provide identifying and entitlement data [6]. To prevent vendors, service providers, etc. from gaining unauthorised access to sensitive information and to facilitate the entity's ability to (1) construct and maintain digital identities for authentication without exposing real identities or relationships between identities, we propose an entity-centric identity management system. The differences between application-centric and entity-centric IDMs are depicted in Figure 2. The data's lawful owner, rather than SP, makes all disclosure decisions about personally identifiable information in an entitycentric IDM. In this study, we provide a procedure for creating an entity-centric IDM model.

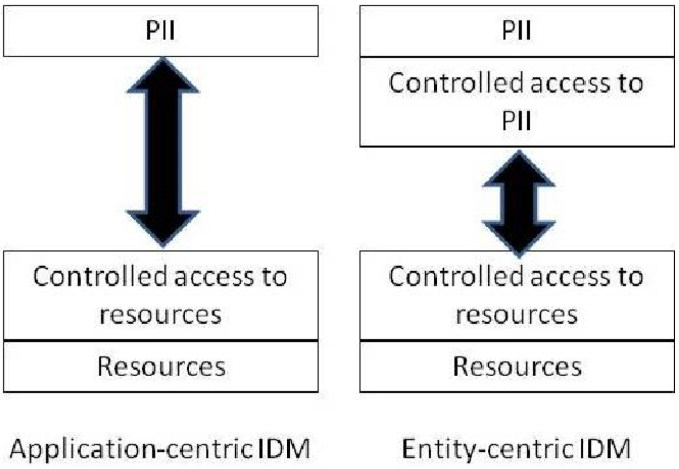

Figure 2. Entity-centric IDM vs. Application-centric IDM

Impact on the Organisation

Our cloud-based approach to IDM is (2) deployable on unknown, untrusted hosts, and (3) independent of trusted third parties. The approach has two main components: (2) Active Bundles, which contain PII, privacy policies, and a virtual machine that enforces the policies and uses a set of safeguarding strategies (such as integrity checks, apoptotic evaporation, decoy) to protect itself; (1) Anonymous Identifiers to mediate interactions between an entity and cloud services using the entity's privacy policies; and (2) Integrity Checks, Apoptotic Evaporation, and Decoys.

The Literature Review

We review three of the available therapies for IDM.

PRIME Privacy and Identity Management for Europe (PRIME)

It provides access to anonymous credentials that preserve users' privacy. The user interface runs the processes for getting claims endorsed by IdPs on behalf of RPs. By using a technique called identity mixing (which is based on the selective disclosure protocol), users can acquire anonymous credentials from an IdP while still having the option to selectively reveal certain characteristics about themselves. The credentials are then digitally signed using a public key infrastructure. PRIME is a barrier to standardisation since it requires both user agents and SPs to adopt the PRIME middleware. [7]

CardSpace (B) for Microsoft Windows

All Windows users' virtual selves The security token for CardSpace [8] is an extension for Internet Explorer 7. A security token is made up of claims, which can be anything from a username and full name to an address and even a Social Security number. The token's owner's assertions can then be checked for veracity.

A CardSpace-enabled website or app will require a user to make a series of claims in order to confirm their identity. Based on the user's chosen InfoCard, CardSpace software communicates with an IdP to obtain a digitally signed XML token containing the requested information, and then sends this token along to the requesting application.

There have been complaints that the CardSpace design puts too much emphasis on a user's opinion of an RP's trustworthiness. Most people either don't notice or don't bother to approve the RP's digital certificate when prompted to do so in order to access the desired website. CardSpace allows users the option of interacting with RPs without certificates. The user's trust in the IdP that issued the certificate is still required even though the RP has supplied a higher assurance certificate. We also anticipate that when a single IdP and many RPs are participating in a single working session, the secure identity metasystem inside the session will rely on a single layer of authentication, namely the authentication of the user to the IdP. If the password is cracked or a working session is hijacked, the entire system is at risk.

Use an Open ID

OpenID [9] is a decentralised authentication system that allows cloud users to create and maintain multiple digital identities, therefore enhancing their privacy and security. An OpenID can replace several login credentials. With this OpenID, she may sign in to a wide variety of services online. She has been able to utilise her own OpenID for logging in thanks to the RP she has been corresponding with. The consumer has previously set up an OpenID with a TTP. OpenID providers authenticate users (often by requiring a password) and then double-check with the user to verify if the RP should be granted access to the user's identity data when an RP locates an OpenID provider. If she is willing to help, she will be sent to the RP along with her credentials so that they may be checked. After the OpenID has been verified, the user is considered to have successfully logged into the RP using the credentials associated with that OpenID.

OpenID has been dubbed "phishing heaven" because to its susceptibility to phishing and social engineering attacks. A common attack vector for credential theft is a website that masquerades as an OpenID provider website [10, 18].

A Review of the Activated Bundle Scheme

An active bundle (AB) is used to transport sensitive data, metadata, and a virtual machine (VM) [11]. Sensitive content has to be protected from unauthorised access, data breaches, incorrect dissemination, and other security risks.

The metadata describes the current bundle and any protections it has. Metadata such as the source, integrity, who has access, who can share, how long it will last, security (including security server id, encryption algorithm used by VM, encrypted pseudo-random number generator, trust server id used to validate trust level and role of host, and trust level threshold required to access), and lifetime can be found in [11], [12]. The enclosed programme is managed and operated by the computer's virtual machine (VM). The fundamental functions of a virtual machine are (a) implementing access control policies for bundles by apoptosis, evaporation, or decoy operations (for example, only exposing to a guardian the data to which they have access), (b) enforcing dissemination policies for bundles, and (c) certifying bundle integrity.

Despite the fact that one of the goals of ABs is to stop malicious hosts from leaking important data, the approach has its own threat model. The most critical need is that hosts successfully run virtual machines.

A Prototype of a Mobile Agent-Based Active Bundle

We have developed a TTP-based prototype within the context of the mobile agent paradigm. The prototype was developed using the Java mobile agent framework JADE [14].

Among the several AB Services available in the system are the Security Services Agent (SSA), Trust Evaluation Agent (TEA), and Audit Services Agents (ASA), as described in Section B.1 of the ABTTP Architecture Description. An AB Coordinator (ABC), AB Destination, DF, and AB are also part of the system.

The components are stored in four separate bundles.

The Coordinator of Dynamic Bundles

ABC operates the yellow pages service Jade Directory Facilitator. It allows agents to add and delete services, modify their profile descriptions, and do targeted service searches. Four agents, AB, SSA, TEA, and ASA, are enlisted in our prototype. Between hosts, they can keep in touch.

Customer Application

ABC, the show's host, is featured in Container 1. Users are encouraged to provide feedback to ABC, which may include sensitive data, metadata, and a proposed move of the AB. Then, it leverages the user's input to build an AB. The AB's qualities emerge from the way in which the AB processes the user's input. The set of tools that make up the AB's virtual machine are also included. When the DF learns of an ABC, it adds the ABC to its database.

ABD - Destination of Active Bundles

ABD serves as a host application for containers. This section's one and only function is to take in active bundles.

Active Bundle Services

Active Bindle Services includes all three of these agents: SSA, TEA, and ASA. The first agent, SSA, maintains a database of AB information. This key is used to encrypt and decode the private information and metadata of ABs. The SSA maintains a database of all the ABs that may be used along with their names, encryption keys, and minimal host trust requirements. The SSA maintains a record on its system with the AB's identifying information. The second agent, the Trust Evaluation Agent (TEA) [15], provides information in response to questions about a host's trustworthiness posed by the Security Support Agent (SSA). The ABs' activities are monitored by ASA, the third agency. It accepts audit data from ABs and saves it in a database for later examination by authorised users (such AB owners or auditors).

According to "Active Bundles," a "AB" is a "mobile agent" that possesses its own unique characteristics and capabilities. It is constructed by an ABC company (for details, see Section B.2.2). When the AB is finished, the owner is given a few options for ABDs to pick from. When the AB finds out where it will be going, it immediately begins getting ready for the move by packing and arranging its belongings. The following step is to develop ABs (for more information, see Section B.2.3). After arriving at its destination, the AB activates on its own (for further details, see Section B.2.4).

What the Bundle's Active Components Do

Here we summarise AB behaviour as a chain of operations beginning with startup and ending with activation.

To get an AB up and running, it needs access control and dissemination control metadata, as well as other sensitive data and information. ABC builds an AB by merging data and metadata and then creating a virtual computer. After this is satisfied, the AB may carry out the rest of the algorithm independently (using its own virtual machine).

Building an AB Structure

One set of keys is used to encrypt the AB, while the other set is used to sign and verify the signature of sensitive content included in the AB; both sets of keys are provided by SSA. Attackers cannot compromise AB's private data and then use AB's public key to re-sign the altered data.

The AB will submit a request to the SSA for the SSA to maintain the AB's security information. Name, decryption key, and minimal trust level at which a host can use the AB make up its identifiers. Protecting a single location where AB decryption keys and related data can be stored is the goal. The AB decryption keys are only given out to approved hosts.

The AB creates a hash value and signs it with the signature key for sensitive data. The signature verifies that the private data was provided by the owner themselves.

AB encrypts sensitive data using the passphrase to prevent unauthorised access.

Bundle activation (B.2.3) As soon as the active bundle reaches the host device, it begins functioning normally (refer to Fig. 3). The steps of the enabling algorithm are as follows: To begin, AB will contact SSA to learn more about AB's security measures and the host's degree of trustworthiness.

In the second place, AB checks if the host's trust level is high enough to provide AB access. Step 3 involves apoptosis in the AB, whereas Step 4 is reached if Step 3 is not met. In the last phase, AB double-checks the security of its confidential data. Confidential information is hashed to determine its value. The validity of the AB is verified by comparing the signed hash value to the calculated hash value. The AB will go through apoptosis if the verification fails (Step 5), but it will decrypt its data securely if it succeeds (Step 6).

AB scrupulously adheres to its privacy standards.

AB reporting back to the host. The ninth phase involves AB sending audit information to ASA. Not only are AB and the host mentioned, but so is the action under scrutiny ("the move to the host").

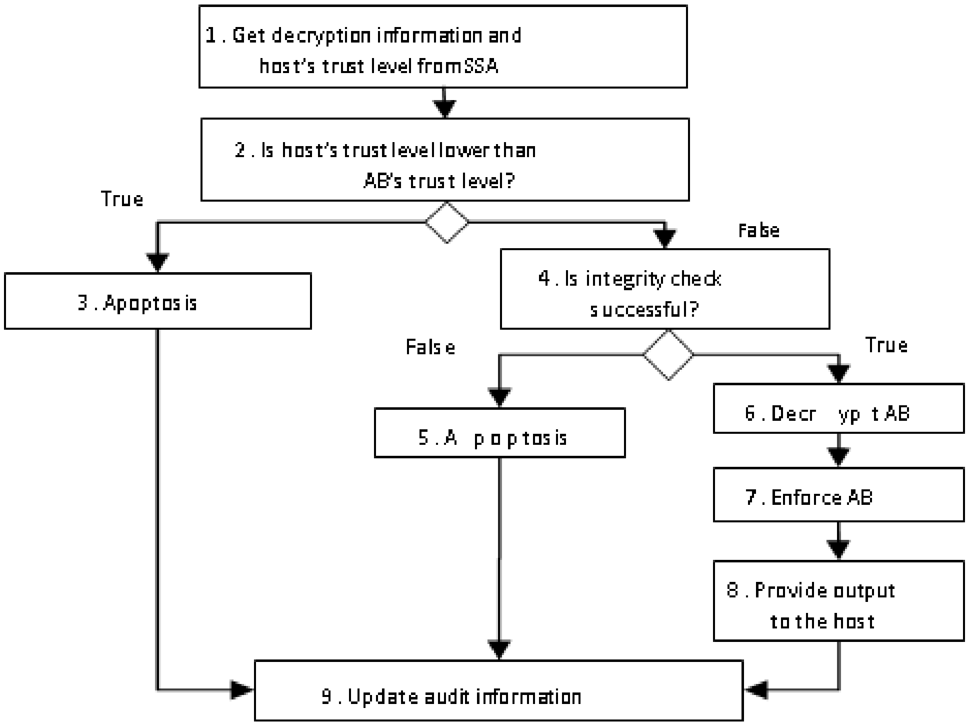
1. Get decryption information and host's trust level from SSA
2. Is host's trust level lower than AB's trust level?
True
False
3. Apoptosis
4. Is integrity check successful?
False
True
5. Apoptosis
6. Decrypt AB
7. Enforce AB
8. Provide output to the host
9. Update audit information

Figure 3. A UML activity diagram to support a bundle's activity.

PROPOSED METHOD FOR SECURE STORAGE OF CLOUD-STORED PERSONAL DATA

In this part, we will discuss the entity-centric IDM paradigm and how the Active Bundle scheme and the anonymous identification method might contribute to it.

A. Characteristics of Existing Approaches

There are two key differences between these solutions:

The first requirement is the presence of a trustworthy mediator. The main issues with this strategy for cloud computing are (i) the service provider and the trusted third party (which may be a cloud service hosted by the cloud provider) might be the same company; and (ii) the cloud provider might not be trustworthy. This means (i) it is a centralised approach, and (ii) the trusted third party may no longer be an impartial authority. However, consumers' private data might be exposed if the TTP is compromised.

Second, untrustworthy servers are off-limits to them. The client application storing the PII must be executed on a trustworthy host to prevent the host from collecting the PII.

Next, we'll look at Picked Study Questions

Among the questions that need answering are:

Having the user's identification data encrypted before being sent to the service in issue is one way to authenticate without disclosing information (encrypted data). However, this data must be encrypted before it can be used by the service provider. Once information has been encrypted, however, it is no longer safe. There are significant privacy implications if the supplier chooses to retain this information.

The present crop of IDM solutions does not permit usage of the service on untrusted hosts (hosts over which the user has no control). Unverified public hosts are not allowed to utilise IDM. Recent advancements in cloud computing have made it possible to store data almost anywhere, making it imperative that this issue be resolved.

Thirdly, security during user-provider communication (against side channel and correlation attacks) is essential. For cloud computing, where one provider may retain sensitive data before transmitting it to another (as a subcontractor) so that the consumer may utilise the service, this is of the utmost importance.

Our suggested solution for entity-centric IDM, in which AB is used to hide sensitive data from hostile sites, is called IDM Wallet, and it corresponds to Strategy C. The method of verifying the identity of a subject without revealing that subject's identity is called "anonymous identification," and it is supported by Zero-knowledge proof. Figure 4 shows the design of the IDM Wallet and how anonymity is maintained.

You can verify an assertion or claim without revealing your identity when you use Anonymous id. However, consider the case of a customer making a book purchase on Amazon. The customer must provide his postal address in order to get the books via mail. Multiple users may be involved in a single transaction, and they will each need their own individual login credentials. Amazon would rather the delivery provider not know the client's address, but it has a responsibility to ensure that the address the consumer has supplied is correct. Upon Anonymous identification, IDM Wallet will create an AB containing the address that must be disclosed. Metadata, access control policies, and virtual machines are also stored in this AB, in addition to the personally identifiable information (here, merely the address). This token is given to SP so that it may be sent on to the post office. Using an IDM wallet protects sensitive data while interacting with untrustworthy websites, while tokens transmitted as AB keep PII protected throughout propagation to SP.

D. An Overview of the IDM Wallet

A user's ability to manage when and where their identities are exposed is stored and managed by their IDM Wallet. The components are as shown in Fig. 4 below.

1) Individually-specific identifiers for purposes of authentication, service delivery, and use, such as social security numbers and dates of birth. This data is encrypted in the IDM Wallet for safety.

Second, the disclosure guidelines Identity information in an IDM wallet must be chosen using these criteria. If a user's credentials have been used in the past to access a certain service, they will need to provide those credentials again each time they access that service. Any other identifying information you provide them is unnecessary.

Third, you may decide what aspects of your Identity to make public based on your previous disclosures. For audits and documentation, this is really helpful.

4) Bargaining strategies: Complete confidentiality during the verification procedure is ensured by the use of Zero Knowledge Proofing. Below, you'll find more information.

Personal information (PII) is protected on untrusted hosts thanks to the programming in the fifth component, the virtual machine. Disclosing rules are strictly adhered to.

E. Recognising Oneself Without Being Recognised Explanation in Great Depth.

Fiat and Shamir's [16] description of their identification and signature method is provided. We discuss the possibility of using this system as a kind of identification.

The Fiat-Shamir Identifier Scheme (E.1) An Explanation

SP may use the Fiat and Shamir identification scheme to verify an individual's claimed identity. SP is not allowed to use the PII of the client organisation to establish its identity [16]. The approach includes both an identity-issuance and an identity-verification mechanism. Before supplying an identity, an IdP chooses a public number n and a pseudo random function f that assigns random strings to numbers in the range [0,n]. Multiplying two unknown primes, p and q, yields the number n.

Entity identifiers issued by an IdP (Protocol 1): After authenticating a user's identity, an IdP will create a string I that includes all of that user's essential information, such as the duration of their identity's validity and any restrictions placed on it. The IdP will next carry out the following steps:

In order to obtain the values of the data at index j, we first use the formula vj = f(I, j).

Determine the smallest square root that contains I, the k values of sj, and their indices, and then select k values of j for which vj is a quadratic residue (mod n).

Publish an identity containing the chosen k indices and the corresponding I, k sj values. (To keep the notation straightforward, we just use the first k indices, j = 1, 2,..., k.)

Assuming n and f are the universal moduli of SP, the second protocol (verification of entity identity) uses these numbers. At this point, Party A must prove to Party B that it is the legal owner of the PII I in question by submitting appropriate documentation to the SP. The steps of a protocol are as follows:

What occurs is that A gives B the information in I. ii) B will supply vj = f(I, j) for every j between 1 and k. Repeat (iii)–(vi) for i = 1,..., t. iii. A picks a random ri in the interval [0,n] and tells B that xi = ri2 (mod n). the binary vector (ei1,..., eik) that B sends to A is fully random. The information sent from A to B is yi = ri eij = 1 sj (mod n).

That xi = yi2 + eij = 1 vj (mod n) is confirmed by B.

In the Fiat and Shamir scheme, the entity (Party A) gives the secret information (I) to the SP (Party B). As a result, B can check if A has an I from IdP. If all the t checks succeed at the conclusion of the protocol, then B will know that I properly identified A, since A does not know function f.

The Anonymity Disclosing Sheme Defined E.2.

We adapt the Fiat and Shamir identification method [16] to prevent Information I from being disclosed from Party A to Party B (as in step i of Protocol 2). The protocol guarantees the anonymity of a value inside a set of values.

In this article, we present a high-level description of a new protocol that SP may utilise to verify entities' identities without accessing their private data. The design incorporates a pair of protocols. Protocol 1 by Fiat and Shamir remains unchanged, whereas Protocol 2 is modified. The fundamental steps of Protocol 2 are as follows: If j = 1,..., n, then IdP will send B the message vj = f(I, j). Repeat (ii) through (v) for i = 1,..., t ii.A chooses a random ri in the range [0,n] and then transmits B xi = ri2 (mod n). iii. B sends A a binary vector with length (ei1,..., eik) that it does not know. The message yi = ri eij = 1 sj (mod n) is sent from A to B.

It is shown by v. B. that xi = yi2 + eij = 1vj (mod n).

In this protocol, every possible attribute value is recorded in SP. It is A's responsibility to locate a correct entry in SP's database that corresponds to the information A possesses.

F. A Proposed Model for Entity-Centric Information Flow Management Implementation

Here's an imaginary scenario where the recommended strategy may be useful:

1) A consumer has a need for a service (a website, for instance) and thus they go to an SP to acquire this need.

2) The SP responds to the service request by informing the entity (through a technical policy requirement) that it must authenticate and provide its identifying information (if necessary) before it may use the service.

The SP's technical policy criteria and the claimant's proof are used by IDM Wallet to choose which claims to issue.

4) IDM Wallet authenticates users with the SP via an interactive protocol based on Anonymous Identification per the entity's instructions, thereby meeting the SP's technical policy criteria for authentication.

5) If an AB token is required, the IDM Wallet will create one and pass it along to the SP. Based on the SP's technical policy, the SP then decides whether or not to recognise (authenticate) the user and provide service.

G. Advantages and Qualities of the Proposed IDM

The proposed strategy includes elements such as these:

To begin, Identity information may be used even on potentially malicious servers. It has a built-in technique that can identify tampered information. In the case that the data's security is compromised, it will obliterate itself by apoptosis or evaporation.

The advantages of the proposed approach include:

Self-reliant and dependable comes first. Since there are no other parties involved outside the SP and the user, 2) Gives as little information as possible to the SP. SP receives just the data he or she requires.

It may be taken everywhere you go. IDM Wallet is portable, so you can take it with you on your phone, USB drive, etc.

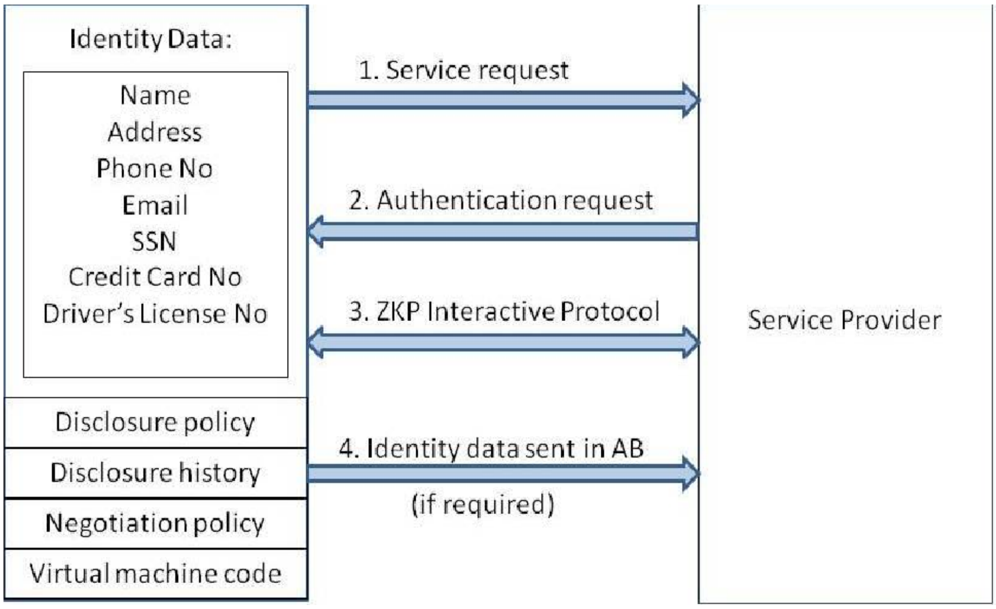

Figure 4. IDM Wallet Architecture

PROTECTING THE PRIVACY OF THE BLIND IS AN EXAMPLE OF ITS APPLICATION.

Those who cannot see have a heightened sense of security and privacy just as the rest of society does. Those who use aids for navigation, such as a white cane, draw unwanted attention to themselves, making them prime candidates for assault.

One emerging design approach [17] for guiding the visually impaired takes use of mobile and cloud technologies to provide context. The proposed architecture is composed of a mobile device with an integrated location sensing module that is responsible for local navigation, local obstacle detection and avoidance, and interacting with the user and the cloud side, and a Web Services Platform that is used to support functionalities like outdoor navigation, indoor navigation, and object recognition.

Tracking a user's or device's location is essential for any context-aware system due to the wealth of data about the immediate surroundings that can be gained from that information. Therefore, location-based services would be among the most often employed in the proposed system for autonomous navigation of the blind and visually handicapped.

If a blind person (or any user) uploads their location data to the cloud, an attacker might theoretically use it to hunt them down and damage or exploit them, raising privacy concerns. Therefore, users' whereabouts shouldn't be tied to the cloud services they utilise. Our proposed identity management system for location-based services follows the minimal data disclosure principle and destroys data through the active bundle when encountering service providers with unsatisfactory trust levels.

Here, we describe the many ways in which the proposed IDM system communicates with an online provider of a pedestrian route planning service.

The user's digital identity includes their name, address, phone number, emergency contact information, and navigation service subscription ID, all of which are stored in the dynamic bundle. By only revealing the subscriber ID and the user's location upon arriving at the service provider, the active bundle is compliant with the minimal data disclosure principle. Before transferring SP sensitive data or a request for driving instructions from point A to point B, the active bundle of the user's mobile device checks with a trust server to ensure SP's legitimacy. If the trust level drops below a specific threshold, the apoptosis function is activated to safeguard the privacy of data stored in an active bundle.

If SP is trusted, the active bundle continues to the next phase, authentication. Disclosure of the subscriber ID at this time would violate the location privacy of the user because it is personally identifiable information. Zero-knowledge proof on the current bundle's virtual machine is used for authentication instead of the subscriber ID. Using this method, we may confirm the user's membership to the service without disclosing any of their personal information. This means the user's privacy is preserved and their whereabouts are undecipherable. Longitudinal behavioural patterns (such commonly visited sites) are much harder to deduce using this approach since a user's identity is not tied to their calls to cloud services.

SUMMARY

Privacy and security have become increasingly important as the usage of cloud services continues to skyrocket in popularity among both governments and enterprises. There is an immediate need for an entity-centric solution to the problem of user privacy protection. Thirdly, it must be able to safeguard users' Personally Identifiable Information (PII), and it must be able to explicitly identify users who can be trusted across the Web and inside enterprises.

Identity management (IDM) is crucial to cloud security and privacy. Cloud computing can benefit from the entity-centric approach to protecting the confidentiality of data throughout its lifecycle. To account for this observation, the active bundle approach was proposed. Users have control over what information is shared and under what circumstances.

After a functional model of the proposed cloud computing system has been developed, it will be put through its paces in a number of realistic scenarios. The goal is to prove that the suggested privacy and identity management framework can function as intended and is suitable for use as a cloud computing industry standard.

REFERENCES

[1] A. Josang and S. Pope. User Centric Identity Management, In Proc. AusCERT, Gold Coast, May 2005.

[2] Wikipedia. Identity Management Systems. July 2010. http://en.wikipedia.org/wiki/Identity_management_systems

[3] K. Cameron and M.B. Jones. Design Rationale behind the Identity Metasystem Architecture. January 2006. http://research.microsoft.com/enus/um/people/mbj/papers

[4] R. Gellman. Privacy in the Clouds: Risks to Privacy and Confidentiality from Cloud. World Privacy Forum, 2009.

[5] A. Gopalakrishnan. Cloud Computing Identity Management. SETLabs Briefings, Vol 7, 2009.

[6] Identity Theft Primer, Libery Alliance Whitepaper, http://www.projectliberty.org/, December 05, 2005.

[7] S. Hubner. PRIME, https://www.prime-project.eu/. 2010.

[8] W. Alrodhan and C. Mitchell. Improving the Security of CardSpace, EURASIP Journal on Info Security Vol. 2009.

[9] OPENID, http://openid.net/, 2010.

[10] K. Cameron, Identity Weblog. 2010. http://www.identityblog.com/?p=685

[11] L. Ben-Othmane and L. Lilien. Protecting Privacy in Sensitive Data Dissemination with Active Bundles. Proc. 7th Annual Conference on Privacy, Security & Trust (PST 2009), Saint John, New Brunswick, Canada, August 2009.

[12] L. Lilien and B. Bhargava. A Scheme for Privacypreserving Data Dissemination. IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, 2006.

[13] L. Ben-Othmane. "Protecting Sensitive Data During Their Life Cycle," Ph.D. Thesis, Western Michigan University, 2010 (in preparation).

[14] F. L. Bellifemine, G. Caire and D. Greenwood. Developing Multi-Agent Systems with JADE, John Wiley & Sons Ltd, West Sussex, England, 2007.

[15] Y. Zhong and B. Bhargava. Using Entropy to Tradeoff Privacy and Trust. SKM, Amherst, NY, Sep. 2004.

[16] A. Fiat and A. Shamir. How to prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO, 1986.

[17] P. Angin, B. Bhargava and S. Helal. A Mobile Cloud Collaborative Traffic Lights Detector for Blind Navigation. 1st MDM International Workshop on Mobile Cloud. 2010.

[18] C. Sample and D. Kelley. Cloud Computing Security: Routing and DNS Threats. 2009. http://www.securitycurve.com/wordpress/