DISTRIBUTED HIERARCHICAL IDENTITY MANAGEMENT: A VISION

Uri Blumenthal, Joshua Haines, Gerald O’Leary
MIT Lincoln Laboratory, 244 Wood St, Lexington, MA 02421 USA
uri@ll.mit.edu, jhaines@ll.mit.edu, oleary@ll.mit.edu

This paper addresses the issues of Global Public Key Infrastructure (PKI). It points out some reasons why PKI has not been as pervasive as it could be, what are its limitations, and what obstacles the current approach places on the road to global usability and interoperability. The paper widens the definition of attributes and proposes putting emphasis on Attribute Certificates. Multiple hierarchical authorities would manage these attributes. The paper brings forth reasons why and how such an approach could be scaled to global community. The paper describes technical and political challenges along this path, and ways to address them. The novelty of this approach is applying architectural design and experience from other Internet sub-systems to Identity Management field.

Introduction

Secure connectivity and trusted data sharing are integral to achieving net-centricity and should manifest as key elements of the net-centric architecture. A crucial part of this is Access Control (also called Authorization) – the ability to decide who should be allowed what. Identity Management (IdM) addresses the who part – the ability to determine and validate the identity of the principal.

Current net-centric architectures provide both Identity Management and Access Control through centralized mechanisms. This approach lacks in three aspects: scalability, resilience and interoperability with various partners (industrial partners for commercial world – foreign in particular, Coalition partners for the military, etc). In order to be usable, Identity Management must include managing attributes\(^2\) of the known entities (see Figure 1). While current approach can\(^3\) deal with bare identities (stripped off many of the attributes needed for Authorization) – its way of dealing with identity attributes does not scale even today. A way to improve the situation and solve the above problems is to employ hierarchical distributed approach to Identity Management.

In addition, the current certificates bundle together multiple attributes, which increases burden on Certification Authorities (CA) that needs to verify all the disjoint attributes of the key holder, and breaks the logical infrastructure of Attribute Authority hierarchy for each individual attribute. Whenever one attribute value changes – the entire certificate has to be revoked, and the whole issuance process re-done. How CA is supposed to find out that an attribute changed is an open question.

Perfect solution – an ideal behavior.

In a perfect environment, the system would

- Retrieve and validate public keys from all over the globe.

---

1 This work is sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2 An attribute is a property or a quality of the principal to whom the given attribute applies. An attribute may be factual (e.g. birth date, height, eye color), or a result of a policy decision (e.g. clearance level). Attributes comprise the necessary input to Access Control logic that in turn renders access decisions. Strictly speaking, “identity” (such as user name, or human name) is also an attribute – that often is used for access control decision (e.g. by employing lists of allowed – or prohibited – principals).

3 We observe that in practice even the “bare minimum” Identity Management interoperability is far behind most every other service or capability deployed on the Internet. The situation in the field – the current practice – is drastically behind the state of the art. Most common identity used today is User ID, authenticated by a password.
- Be able to function – at some increased risk – if online certificate status verification is not available (e.g. when an OCSP\(^4\) responder is not reachable, or a CRL\(^5\) is not available).

- Retrieve and validate attributes from multiple sources all over the globe. Must be able to function when some or all attributes either are unavailable, or their status can’t be verified.

- Be able to assign a degree of trust to each of the keys and attributes it receives.

- Arrive at an Access Control decision by computation using as input:
  a. all or some of the relevant attributes and their corresponding trust levels;
  b. a policy that specifies an acceptable degree of access risk.

These computations must deal with probabilities and risk/trust levels. In each case, the policy defines at what point the risk becomes unacceptably high (e.g. insufficient number of required attributes with acceptable level of trust). But in no case can the system freeze or become unable to give an answer. Such a system must be both distributed and localized. This would allow the “leaf” enclaves to function even when they are cut off the “mainland” of the ‘Net, and the ‘Net – to survive disappearance of any “leaf” enclave.

A centralized system is likely to survive injuries to the leaves, but would collapse if the hub is successfully targeted or disconnected (e.g. by a successful Denial of Service attack). A distributed system should exhibit great resiliency – as evidenced by the example of Internet itself.

**Figure 1. Role of attributes in Access Control.**

**Issues.**

Access Control needs validated identity of the principal that requests service. This identity is comprised of various attributes, some of which Access Control uses. This set of attributes must somehow arrive at the Policy Decision point. Once the attributes are retrieved, their level of trustworthiness (numeric representation) must be determined. The issues are:

a. quality of the attributes (their trustworthiness)\(^6\).

---


\(^5\) Certificate Revocation List – a list of certificates (or usually just their serial numbers) that have been revoked or are no longer valid, and therefore should not be relied upon. One needs mechanisms like OCSP and CRL to deal with cases when a certificate becomes invalid prior to its expiration date.

\(^6\) Related to quality is the property of cost – i.e. how much does it cost to the issuer to ascertain the correctness of the data he’s vouching for. Such cost should be lower when the authority and responsibility are distributed along the hierarchies -- so that each entity vouches only for what is under its direct control.
b. *availability* of the attributes (their authoritative source may be unreachable, or unwilling to share the data).

Traditional approach based on Risk Avoidance has been "if I certified this attribute – then it’s 100% trustworthy; and if it comes from outside (whatever that means) I don’t trust it at all". To interoperate with people outside your organization, this approach has to change to Risk Management. It requires ability to assign a digital representation of risk or trustworthiness to authorities outside of your control, and take that number into account in the policy computation.

**Current situation.**

Existing architectures in the industry and military alike consider its “communications universe” as one huge enterprise, like one “corporation”. There are two main aspects that characterize this “enterprise” and its network:

- Enterprise has complete *authority* and control over all the nodes comprising their network. Enterprises usually are centrally governed, and they can both define a policy and enforce it on their networks (if and when they have a will to do so). Violators can be prosecuted – this is both a deterrence factor and a way to recoup the damage via litigation (which for a business usually is good enough).

- Enterprise networks are not built to *withstand attacks* by adversaries of equal or greater resources and power – like a typical police force is not designed to protect a town from a military attack.

The Coalition military environment makes both of these presumptions untrue.

First – even in the military it is not easy to consistently implement a single policy spanning from top to bottom all the branches and services. To complicate the issue, there are partners residing in jurisdictions outside of the principal’s government authority. Yet there is a need to communicate and interoperate, which in turn requires establishing and validating peer identities, and – which is even harder – deciding on the fly what access rights to grant to that identity (and for how long). For example, US Army and US Navy could agree on what identity attributes of their personnel they would disclose to Joint Enterprise Directory Service (JEDS). But a US ally may not wish to send certain identity attributes of its personnel to US. Where does this leave occasional coalition partners, such as Syria, Saudi Arabia, etc. etc?

Second – centralized infrastructure even in “good” times exhibits two weaknesses: bottlenecks and single point of failure. Combat situations demand (reasonable) autonomy of the nodes, who often find themselves either isolated from the central authority or connected via “austere” link that does not support high-volume traffic necessary to support centralized identity management operations. Thus to disrupt this infrastructure, an adversary would need to either bring down the central “node” (for example JEDS), or disrupt the communications between the central node and the local (theater, tactical, etc) entities. In these conditions each node has to either maintain its own copy of a potentially huge database that lists everybody who may communicate with it, or drop requests when the central “clearinghouse” is unavailable. Centralized approaches can barely scale to the world’s size in time of peace; it can’t be anything but a disaster in time of war where adversaries will do all in their power to disrupt our communications. The principal does not have a choice of who to present his individual attributes – it is a wholesale deal: either you give your certificate with everything that’s in it – or you don’t present your certificate at all.

To summarize, the weaknesses of the centralized identity management architecture as it relates to identity attributes are:

---

7 Problematic to maintain, and unacceptable in view of possible security exposures.

8 There is no existing example of a functioning global “Big Brother”. Companies like Google or Amazon.com do manage huge networks and databases - but they are not facing determined attacks by cyber troops of national states, and despite that friendly environment they have their share of access failures.
Many entities that one needs to interoperate with will not be in the “central” dossier database regardless of how large that database is going to become.

A huge central identity attributes database is a single point of failure that is (a) difficult to protect adequately because it has to be visible and accessible by everybody (not just “by many”), and (b) cannot be protected by sufficient replication because of sensitivity of its contents. As an attempt to remedy this situation, vendors added configurability to their products – each node can manually add several “authority points” and go to the next one when the main point is unavailable. It is an improvement over the existing situation – but is hardly a solution because (a) it requires too much manual intervention, and (b) it still lacks structure similar to that of military chain of command.

We view Identity Management in a wider context of authentication of the principal and making authorization decision on the request (because usually we authenticate in order to decide what to allow). We observe that in real world both authority and identity control usually are distributed and hierarchical. For an example of time-proven hierarchical scalable architecture we can turn to Internet. Its hierarchical decentralized design proved working in world’s scale. This scalability and hierarchy can (and should) be applied to identity management as well. In particular Internet Domain Name Service (DNS) can serve as a good example of a scalable and resilient approach that survives even when parts of the tree are temporarily unavailable.

Proposed Approach

9 The more “replicas” of such a database are deployed – the harder it is for adversaries to bring it down, but at the same time the likelier for adversaries to obtain its contents from at least one of the copies.

10 E.g. platoon leader knows his platoon, battalion commander knows his company commanders but probably not every soldier in every platoon, etc. If permission is sought at any level – it is either decided right on the spot (locally), or elevated one level.

Figure 2 Current use of PKI

For an example of time-proven hierarchical scalable architecture we can turn to Internet. Its hierarchical decentralized design proved working in world’s scale. This scalability and hierarchy can (and should) be applied to identity management as well. In particular Internet Domain Name Service (DNS) can serve as a good example of a scalable and resilient approach that survives even when parts of the tree are temporarily unavailable.
Certificates vs. real-time Security Assertions.

In general there are two ways of dealing with credentials associated with service requests:

1. **Security Assertion**: at the time of the request, the verifier asks the issuer what credentials\(^\text{11}\) the requester has, or

2. **Certificate\(^\text{12}\)**: at the time of the request the requester presents to the verifier his credential that an issuer gave him some time ago.

Advantage of the first approach – using Security Assertions – is that the answer is always fresh, while its disadvantage is that it requires that the issuer is online and accessible (via sufficiently wide bandwidth) *at the time of request*. This disadvantage can be very severe for tactical, disjoint or isolated, and for austere communications.

Advantage of the second approach – using Certificates – is that once the credential has been issued, the requester can operate “autonomously” without the need for the issuer to be always present online, as the validity of the credential can be ascertained without direct contact with the issuer. Its disadvantage is that requires an extra mechanism to deal with the possibility that the authorization from the requester has been withdrawn\(^\text{13}\) prior to credential’s expiration. The solution currently used is Certificate Revocation List (CRL) that lists certificates, which have not expired, but for some reason are not valid any more.

CRL seemingly nullifies one benefit of Certificates – its ability to operate without direct real-time communications with the issuer, because now instead of asking the issuer whether a given principal has a given property, one has to contact the issuer and ask whether the given property certificate is still valid. There are two important differences:

1. CRL does not include any information about principals whose credentials are revoked - only serial numbers of those credentials.

2. CRL are issued periodically and – because they are not sensitive to disclosure – can be replicated and propagated through the entire network. To reduce traffic load, instead of transmitting the entire CRL every time, only the changes are sent (called deltas). This way even an entity with austere connection can make use of CRL. And even if only an older copy of the CRL is locally available – the policy can account for the fact that there is a certified key, which has not been revoked as of (e.g.) yesterday.

This is how the “main” identity is dealt with today. But to make access control decision, it is not enough to know who the principal is – his attributes are required as well. The peer must know where to get values of those attributes and how to validate them.

One valid question is whether certificates are more susceptible to capture, spoofing or exploitation by non-intended users. The main difference is: SAML assertions are usually sent by the “issuer” and are short-lived (so the principal and third parties would not even see it, much less able to intercept and exploit later), while attribute certificates are usually possessed by and sent from the requester himself. But there is no reason why certificates can’t be awarded the same degree of protection as SAML assertions, e.g. transferred over an encrypted channel to avoid capture by eavesdroppers. So the only remaining issue to consider is whether the requester is allowed to possess his own “certified attributes” or not. The proposed certificate-based approach would not suffer or change if some of the attribute certificates were to be obtained from sources other than the requester himself.

---

\(^{11}\) Security Assertions usually are encoded in Security Assertion Markup Language (SAML), and called SAML assertions.

\(^{12}\) A “certificate” is basically a signature of somebody of authority over the principal’s Public Key, stating that “I attest that this key belongs to so-and-so and is valid from AA/BB/CCCC vv:zz to DD/EE/FFFF xx:yy”. Current state of the art for identity validation both in the military and in the industry around the world is via Public Key Infrastructure. A principal is issued a Public Key pair that is certified. That “proves” his identity when it needs validation.

\(^{13}\) For example, when a person is fired, or loses his clearance; or his key is stolen and must be canceled immediately.
We must admit that it is possible to use specially crafted\textsuperscript{14} and issued certificates in place of SAML assertions, and SAML assertions in place of long-term certificates. But while technically possible, it seems counter-productive. On the balance – certificates win.

\textbf{CRL and Attribute Authorities.}

Certificate Revocation Lists are already pushing the limits of practicality, forcing alternative mechanisms such as online certificate status verification (which is good from the client’s point of view, but it negates the main advantage of a certificate over an online assertion) and CRL deltas (sending only the changes to the previously issued CRL).

Thus a legitimate question is “Aren’t we exacerbating the already-bad situation with CRL size and complexity by adding more authorities with more need for CRLs?” The answer is – Attribute Authorities and Attribute hierarchies will improve the situation by relieving the “main” Certificate Authorities from the need to invalidate the “main” certificates quite so often. Current approach of cramming issued certificates with multiple attributes causes the need to invalidate a certificate whenever any of the attributes becomes invalid. As most of certificates are issued by one of the few Authorities, CRLs that they produce, end up being quite large. Spreading responsibility among multiple authorities, and the attributes – between various certificates allows invalidating only the relevant certificate that would impact only one CRL. Verifiers that are not interested in that attribute won’t even bother retrieving it. And each individual CRL would be considerably smaller.

\textbf{Characteristics of Distributed Hierarchical approach}

Data owners maintain and validate their own data. No unnecessary replication takes place, eliminating the extra possibility for the data to become “stale”. When a particular record is needed, the owner is asked for it, and he provides the answer based on his local policy. The novelty (compared to other existing projects in the Net-Centric sphere) is in defining hierarchic structures of identity authorities, and attribute authorities for each attribute.

The requester’s policy will take into account that (a) not all the data owners will share the data with him, and (b) not all the data owners are equally trustworthy.

The main disadvantage is that today there is no established infrastructure to deal with attributes.

Central repositories represent an attempt to side-step the need for such an infrastructure. “You need to validate identity – just go to VeriSign. You need somebody’s attribute – just go to JEDS.”\textsuperscript{15} Unfortunately, infrastructure must be built for both entity authentication and identity management. Just like if there were no gas stations, one still couldn’t deliver all the gasoline to D.C. and have all the drivers go there to fill their tank. At some point this approach breaks\textsuperscript{16}.

An important issue is what are the assumptions about online presence of various attribute authorities – i.e. whether our approach demands that all the relevant attribute authorities are always online. This translates to the question of whether we can assume that if a requester from an organization X has connectivity and asks for services, then “his” attribute authority also has connectivity and will be able to vouch for him by providing his attributes. Attribute certificates alleviate this concern, as only Attribute CRL need to be available online – which can be provided by e.g. Akamai.

Infrastructure requires effort and delegation of responsibility. One would have to:

\textsuperscript{14} One can issue a certificate with a very short lifetime to imitate assertion’s short life span. One can invent, add and encode extra fields in assertions, including assertion serial number – and then keep track of those assertions, publish CRL, etc. Not that such a practice would make sense.

\textsuperscript{15} The existing architecture does allow for the service to ask other attribute repositories. The problem that remains unsolved – the service must know in advance what repositories to ask. It breaks completely the “unanticipated user access” model.

\textsuperscript{16} Supply system in Soviet Russia worked in a similar way: most of the goods from all the USSR were delivered to Moscow, and if you needed it badly enough – you’d take a trip there and try to procure it there.
a. Assign a certain degree of trust to each class of “attribute authorities” – and maybe to individual authorities within each class, and

b. Craft local policies that accommodate varying trust levels. This is Risk Management instead of Risk Avoidance approach.

Figure 3 depicts the components of the proposed architecture. A principal has a (certified) public key pair (Main Key Stem). In addition, he has several attributes in the form of Attribute Certificates that are tied to his public key. For each attribute there is an issuer. For example, the attribute Employment could be issued by the principal’s employer personnel department. Each Attribute Certificate issuer in turn is certified to operate by a Certifying Authority for this given well-known attribute. Certifying Authorities would verify issuer’s compliance with the governing policy and review issuer’s audit reports. Certifying Authorities are certified by Root Authorities for a given attribute – which provide governance, publish policies and compliance requirements, verify Certifying Authorities’ compliance and examine their audits. Unlike traditional identity certificate, Main Key Stem contains only the minimum number of attributes (ideally none – but it would not be practical), such as key serial number, key expiration date, etc.

Verifier – the entity that needs to make an access decision for this principal – receives principal’s signed public key and attribute certificates only for those attributes that it needs in the decision making process (other attributes are not relevant and therefore not delivered to the verifier). Verifier obtains Access Control policy from Policy Authority (whatever that means in the given context), and applies it to the attributes he got from the principal. In the given example the access policy requires a certain value of Attribute B, so the verifier requests and the principal provides his Certificate for Attribute B. Verifier validates the main key (including expiration and revocation), checks that the given attribute is for the provided key, validates the attribute certificate itself (signatures, expiration, revocation by the issuer) – and finally examines its value. Then if the policy tells to allow access for holders of this value, the access is granted – otherwise it is denied.

Attribute Management Infrastructure.

For identity attributes we envision the following requirements:
1. A high central authority\textsuperscript{17} issues a policy defining and supporting creation of Attribute Authority hierarchy.

2. Attributes will be issued at the local places where the actual knowledge resides – for example, employment attribute would be provided by employer’s HR department.

3. Attribute issuer would be in a certified chain specified and approved by the central authority.

4. Compliance with certification policies would be mandatory, and auditable – similar to the process established in the financial\textsuperscript{18} industry for setting policies and verifying policy compliance. Compliance audits would be performed by the government and by authorized private companies such as Deloitte – like SAS-70\textsuperscript{19} compliance validation in the financial industry. Compliance validation reports will influence the degree of trust placed on a given Attribute Issuer.

Requirements for an Attribute Certificate issuer:

1. Each organization assigns a management group (starting at a certain level) that keeps track of its personnel attributes.

2. Each management group delegates a point of contact that serves as a responder to attribute queries from both local and remote requesters.

3. Each management group must abide by a local policy provided by the organization as to how to deal with requests from various classes of requesters.

4. Each management group knows point of contact of each of its subordinates.

5. Each management group knows point of contact of its superior.

6. Identity must be accompanied by an explicit attribute defining principal’s position in the hierarchy (e.g. 4th Brigade of Royal Marines) and the initial point of contact for this principal’s attributes.

There are political and technical components of this problem.

Issues

Political issues.

It requires considerable effort to establish a distributed hierarchy system, especially so when a separate hierarchy is required for each one of the attributes. Common sense suggests there won’t be too many attributes in use, but the useful set is likely to be in the ballpark of 20 to 50 attributes. That means we may need to establish, maintain and regulate 20 to 50 separate distributed hierarchies. As Return on Investment is not likely to appear until there has been significant deployment, it is improbable that the industry would take the lead. The only feasible way to create such infrastructure in realistic time frame is for the government (or probably governments) to sponsor and support this project.

For the industry to be able to reap benefits of this approach, the government must provide an adequate framework, including:

\begin{itemize}
  \item An authority with the power to issue binding policies, rules and regulations. Government would satisfy all these requirements.
  \item While it is true that financial industry is not perfect, and several crooks managed to find and exploit loopholes to the detriment of many – one cannot deny its successful operation for many years and considerable accomplishments in global interoperability. So rather than pointing fingers at what’s wrong there, we could learn instead what they did right and bring the benefits of that experience into rather less advanced field of Identity Management.
  \item SAS-70 is \textit{Statement on Auditing Standards (SAS) №70}, a widely recognized auditing standard developed by the American Institute of Certified Public Accountants (AICPA).
\end{itemize}
- Legislation and policies. Government should define authority and responsibility of the regular organizations (private and public corporations) that would have to shoulder another burden – managing certifying authorities that issue certificates for the attributes they have control of (such as employment verification). These regulations would also deal with outsourcing this functionality. There already are precedents of government mandating certain things and actions in financial area – bookkeeping, and overseeing authority – U.S. Securities Exchange Commission (SEC). A similar overseeing body would be necessary for:

- Compliance verification and auditing. A body like SEC would do the oversight, with private companies doing the bulk of work, with a defined formalized process and output (such as SAS-70).

- Government sponsorship – to compensate the “pioneers” for the expenses they’d have participating in this project.

- Standards – agreeing among the participants what the useful attributes should be, how they should be expressed and encoded, what the validation requirements are.

Technical issues.

The hard parts of this problem are:

- Quantifying the risk and assigning “trust” or “reliability” numbers to various Attribute Authorities.

- Designing mathematical algorithms that incorporate those numbers into their access decision computations. Such decisions are likely to be expressed as numbers.

- Designing policies that operate with degrees and levels of risk, making acceptable decisions based on the received values and probabilities of their correctness, including Situation Awareness – also expressed as a set of attributes – into consideration.

This would reflect a major shift from Risk Aversion approach to Risk Management, which is not simple in either technical, or political sense.

Related work

The most prominent project that defines attributes and is the biggest consumer of those is Net-Centric Enterprise Services (NCES). Several other projects utilize NCES services and share its IdM approach. The issue and the need-to-do as we see it, is:

1. Cleanly separating “identity” from attributes (name, rank, personal number, etc. are all examples of attributes that do not belong together with the main “identity” token).

2. Creating an infrastructure for attribute authorities for each of the useful attributes (it’s hard – so understandably people have been trying to design around it rather than tackling this issue).

3. Within the above infrastructure(s) creating automatic transitive attribute-fetching mechanism. How does it differ from NCES’s proposed “attribute discovery” – currently there is no hierarchy among the attribute “authorities”, so the task of determining who the appropriate authority is for a needed attribute of a given identity does not have a straightforward solution (i.e. “discovery service” is bound to have sweet time chasing attribute authorities, since there is neither a hierarchy nor an established “blessed” relationship among them). Especially considering that for different

---

20 Trust or reliability in the context is related to and can be expressed (and used in computations) as probability. For example, reliability of 0.9 would mean that there is 0.9 probability that the received value is genuine (or trustworthy, depending on the context).
attributes (of the same principal) there are likely to be different authorities. Without the ability to automatically span the (group of) tree(s) of attribute authorities to locate the needed attribute for the principal in question, both scalability and ability to deal with unanticipated users would be diminished.

Reviewing the NCES approach actually "gave birth" to our concern and our proposed solution. It (NCES) could theoretically utilize a hierarchical approach, but (a) it isn't considered now as far as we know, (b) all the NCES-related implementations we know of don't support that mode of operations (and there are no plans, as far as we know), and finally (c) there is neither the infrastructure for identity attributes nor any attempt to create it.

Summary

We analyzed the current state of affairs and the perceived needs of the Identity Management in the framework of Net-Centric Operations. We outlined the problems that prevent wide interoperability and scalability of various PKI users and infrastructures. We described steps and measures that would create and support such interoperability. We described technical and political issues that complicate implementing our proposal, and pointed out that political issues dominate over technical ones here, and governmental support can turn the tide.

Acknowledgements

The authors are indebted to Philip Minor (DoD OSD-NII) and Andy Bentley (MIT Lincoln Laboratory) for many helpful comments, reviews and suggestions. The authors are grateful to Bruce Benjamin from JHU APL for a productive discussion.

Bibliography


5. American Institute of Certified Public Accountants (AICPA) http://www.aicpa.org/

6. SAS-70 audit developed by AICPA http://www.sas70.com/about.htm


