A Reference Architecture for the Jericho World

P. A. Galwas and A. Peck
nCipher Corporation Ltd.,
Jupiter House, Station Road, Cambridge, CB2 2JD,UK

paul@ncipher.com

Version: 31 May 2005

Abstract: Safety in a de-perimeterized world is predicated upon mutual authentication and confidentiality: define and verify whom you trust - then grant them alone the rights to access your data and processes: identify, protect & control.

This paper outlines a reference architecture to achieve this now, and outlines a road map showing the key steps towards deployment, ultimately culminating in the nirvana of cryptographically-based, fine-grained protection.

It enumerates the four critical architectural building blocks and the four supporting open standards; applies them in the context of existing system components and highlights a migration path to leverage emerging technologies.

By concentrating on this minimal set of standard components your corporation can achieve the business benefits of scaleable de-parameterization, while optimizing flexibility and capital & working expenditure.

1 Introduction

User-led industry groups such as the Jericho Forum[1] acknowledge that traditional security measures such as firewalls need to evolve towards a new, de-perimeterized world:-

"Trust and verify. Establish those whom you trust. Verify that they are who they say they are. Make sure they only have access to data they need. Ignore everything else. Do that, and you can extend your business as fast as you can set up an IP session and blast **encrypted data** across it."[2]

The de-perimeterized corporate IT infrastructure needs to *Identify, Protect and Control*:

- *identify* a specific person, or application or hardware device - or their combination - and the systems that they are accessing;
- *protect* data and *control* its confidentiality using:
  channel security - to communicate between mutually authenticated parties;
  content security – to protect the confidentiality of long-lived data wherever it may be stored - in a file, database or archive medium.

However, unless an enterprise addresses these requirements in uniform and scaleable ways, the costs of implementing and managing the resulting system will be prohibitive.

This paper proposes a minimalist system design that can meet this challenge now using existing standards and products, while providing a platform for the next stage of evolution: to augment *control* to provide and protect fine-grained access to specific functions in computer processes and accessing data[3].

---

[1] http://www.opengroup.org/jericho/

[2] http://comment.zdnet.co.uk/other/0,39020682,39173924,00.htm

[3] See Moulds & van Someren's Jericho Challenge paper

Section 2 outlines the business drivers, which frame the high-level requirements for the building blocks described in Section 3.

Section 4 describes a reference architecture for the Jericho World from these building blocks and Section 5 relates the architecture to the status of industry standards.

Section 6 summarizes the natural phases for an enterprise rolling-out this reference design.

2 Business Drivers

De-perimeterization offers the promise to cut capital and operational IT costs, for example, by allowing high, fixed network costs to be replaced by on-demand networks. However to achieve this, corporations must tame the costs of scaleable security[4].

Mutual authentication of people and computers is needed, but deploying security measures across thousands of users and hundreds of diverse applications have traditionally resulted in unacceptable complexity and cost.

Scalability and cost are ultimately what count – provisioning identity and enforcing security policy across thousands of end-points requires automated systems that are scalable in order to reduce complexity and cost. This reference design is driven by these core business drivers:

- Reuse of components to protect existing investment
- A minimal set of core building blocks, based on industry standard interfaces, to unify the approach
- Flexible migration from legacy to new components and approaches
- Centralize policy enforcement and automation, to allow scaleable management and audit.

3 Building Blocks

3.1 Identify

Each actor (person, computer, program, data element, peripheral, resource) in the de-perimeterized world must ultimately have a unique identity – its *name* - wherever it might be.

In the case of a person or program, this identity is associated with a *credential* to establish who or what seeks access to some resource. This is well-understood for closed communities – although there are future challenges for cases of more loosely-formed federations, which are outside the scope for this paper.

This identity is also the glue that binds relationships between resources - and this is less widely appreciated.

One example is when an Internet user is authenticated at the first point of contact with the system, and that authenticated state must be communicated to distributed downstream processes. Another example is where one needs a robust association between a piece of encrypted data and the identity of the cryptographic key to decrypt it.

To be effective, an identity must be independent of the various implementation mechanisms, such as a database that stores the associated resource and must apply in a uniform, open and scaleable way across the enterprise.

Moreover, in the de-perimeterized world, one must secure these associations between resources against corruption, for example to prevent someone maliciously changing the policy by which a resource is processed.

---

[4] "Security in an Island World", KPMG & nCipher, 2004, http://www.ncipher.com/resources/islands-wp/

Cryptographic credentials are the method of choice for achieving these ends, for example by using suitably protected cryptographic keys to underpin identity, and digital signatures to enforce the authenticity and integrity of associations. This view is reflected in the widespread use of public key cryptography to support authentication and integrity in modern standards.

However, it is also necessary to provide a flexible migration path that integrates numerous legacy credentials, such as passwords and tokens, with cryptographic credentials.

To unlock this dilemma, one can optionally allow the independent authentication of one or more legacy credentials to access the primary cryptographic one. This approach is already widely used for smartcards, where a PIN is needed to unlock identity keys on the card. Separating legacy credentials from the cryptographic identities that underpin authentication must be recognized as a universal architectural principle. Scalability cannot be achieved, unless the combinatorial interplay of the different legacy credentials is removed from the applications, whose use the credentials protect, and centralized near the point of control of the cryptographic credentials.

3.2 Protect Channel

A *secure channel* protects data 'in flight' over insecure networks such as the Internet, by providing confidentiality of data and by ensuring that data is sent only from suitably authenticated sources to authenticated recipients.

The Virtual Private Network (VPN) is one established approach to channel security, but experience shows that VPNs do not scale well, since they result in a combinatorial increase in the number of channels between parties – making management complex- and they handle privacy and authentication at a low level in the network stack too far away from business processes to allow fine-grained control.

In contrast, client-authenticated Secure Socket Layer (SSL) - and its derivatives such as SSH and 802.1x-EAP/TLS - allow specific authenticated sessions between two parties (a client and server) to transfer some specific data between them in the context of a specific business process.

3.3 Protect Content

Secure content mechanisms protect important parts of a resource from creation, 'at rest' when stored in a database, over the networks on which it passes, right to the point where it is used.
There are three distinct aspects to content protection:

- Protecting against the breach of confidentiality that results when data is stolen from storage
- Allowing control over who or what may access given data – and from where – for example by regulating the availability of the associated decryption key
- Allowing fine-grained access control – who may do what with the data - to achieve sophisticated digital rights management.

XML is emerging as the standard for representing data as it flows between core business processes. It is increasingly the preferred approach to unifying data transfer and interoperation between legacy applications; it underpins the Web Services architecture and an ever-increasing set of standards for specific business domains. The XML standards provide for encrypting and signing such data.

However, it is also necessary to consider protection of data in files and relational databases.

Traditional operating system access-control mechanisms traditionally provide protection, but do not offer a unified or standard way to integrate with other control mechanisms.

Increasingly disk-level, file-level and field-level encryption are being deployed.

3.4 Control

There are two principal points of control:

- authenticating an actor before authorizing an operation, for example when establishing a secure channel or executing a specific business function;
- granting an actor access to encrypted data, by giving access to the decryption key.

The supporting mechanisms must scale, which typically requires group-based membership semantics to tame complexity, and centralized management of the control policies.

From a security perspective, enrolling an entity is critical and frequently must have a manual component; revoking in a time-bounded manner is essential; and resilient automation is needed wherever possible.

A traditional PKI supports for certificate-based authentication, typically using OCSP to bound revocation to an acceptable limit. However, increasingly within closed domains, there is a move from traditional black-list PKIs to white-list ones, which only store valid public keys and remove invalid or expired ones immediately on revocation.

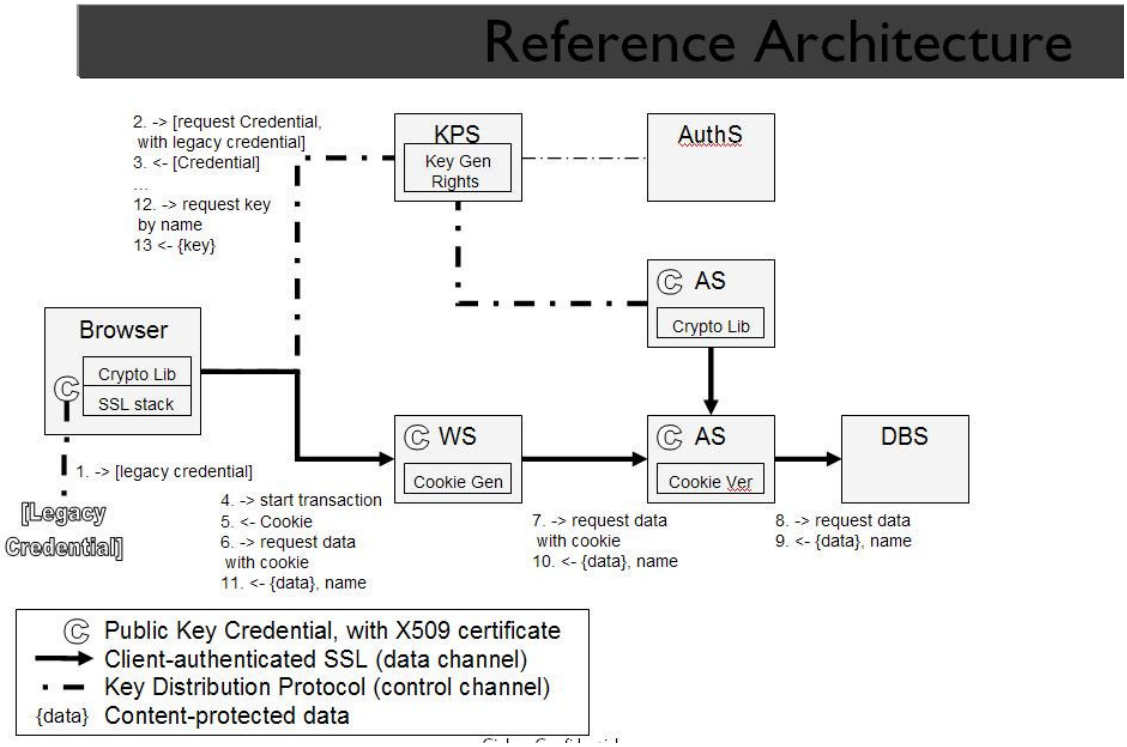
To decrypt long-lived data one requires the private (or secret) key. Without revocation of the private key - to prevent decryption of confidential data - it is not possible to control protected content. Therefore, the decryption keys must be delivered under control to the actors who may use them, with sufficient reliability to ensure that business operations are not compromised, including a comprehensive escrow framework.

We call this capability a *Key Provisioning System* (KPS).

A KPS is also well-suited to controlling distribution of public keys to support white-list authentication schemes and maintaining and securing the mapping between an identity's name and the associated cryptographic keys that underpin its electronic identity.

4 Reference Architecture

The figure 'Reference Architecture' shows the essence of the reference architecture.



An identity is represented by a name and uses a public key-pair as its credential, C. When it communicates over SSL it has an associated X509 digital certificate. Standard SSL stacks, such as OpenSSL and Microsoft Windows support this functionality and certificates can be obtained from standard certificate authority products or services.

When a legacy credential is also used, it is validated either in a local operation to grant access to the private key; or to the KPS which returns the corresponding PK credential only after validating against a legacy authentication server, AuthS.

Industry standard browsers can communicate over client-authenticated SSL with standard Web Servers, WS; and client-authenticated SSL provides channel security between a Web Server and an Application server, and an Application Server, AS, and Database Server, DBS. These SSL servers may be customized to use OCSP to validate client certificates to reduce revocation lag; or to use a white-list methodology.

Whenever a browser begins a transaction at a Web Server, the a *cookie generator* constructs a SAML cookie that is signed with a domain identity, which can flow around the system to be checked by a *cookie verifier* in back-end Servers as required: it becomes a proxy of the browser's, or another program's, original credentials.

Content-protected data is encrypted for confidentiality using a named key, supplied by the KPS. The data contains the name of the associated decryption key. For example, the encrypted fields in a database record contain a field that names the decryption key; encrypted XML data contains a field with the key's name.

When an application receives content-protected data, it requests the key of the given name from the KPS. The KPS provides the decryption key, provided that the requester may read the content.

Standard cryptographic libraries, such as MS CAPI, PKCS#11, OpenSSL, can be modified to transparently receive keys from a KPS.

5 Standards

The Public Key Infrastructure (PKIX) standard suite is well established under IEFT[5], using asymmetric cryptography for credentials and X.509 for digital certificates.

Secure Socket Layer (SSL) has long been established as one of the primary users of public-key credentials, and has been developed more recently under IEFT[6] as Transport Layer Security (TLS). TLS is being used widely for authenticated channel security, for example, 802.1x EAP/TLS[7] for port-level access control in wireless LAN and elsewhere, and is widely recommended to secure XML standards,

Security Assertion Markup Language (SAML) is a more recent, but stable standard developed by OASIS that has extended PKIX to support PKIX authentication objects in the context of XML[8] and can readily be packaged as a cookie (for example see Boeing's deployment[9]). The W3 XML Digital Signature Group[10] has developed standards for encrypting and signing XML data, which form a complete and stable set of building blocks for cryptographic primitives that are needed for content protection.

There are currently no standards for private and secret key provisioning, however in due course nCipher intends to make its Key Distribution Protocol (KDP) available in the public domain. To date, nCipher has licensed the KDP to various business partners. Specifically Broadcom have announced licensing this for their BroadSafe technology[11] and have recently announced the first commercial chip to utilize KDP - an all-in-one VoIP telephone chip[12]. KDP is complementary to existing PKI standards and sits alongside XKMS, which predominantly provisions digital certificates[13].

The one area of the reference architecture that is not currently standard-based concerns content security for data in operating system files and relational databases. While pragmatic customizations, using industry standard cryptographic APIs, such as MS CAPI and PKCS#11, can implement these capabilities now, perhaps Jericho could encourage standardization in these areas.

6 Roadmap

There is a natural roadmap to the deployment of this reference design: the essential steps are:

1. introduce and grow public key credentials as the uniform authentication means - cutting costs of authentication;
2. roll-out mutual authentication, using SSL channel security – enabling a controlled community of actors (people and computers) who may access the corporate systems;
3. decouple legacy authentication from specific application, by using a KPS – providing a flexible and scaleable route towards deployment of strong credentials;

---

[5] http://www.ietf.org/html.charters/pkix-charter.html

[6] http://www.ietf.org/html.charters/tls-charter.html

[7] http://www.faqs.org/rfcs/rfc2716.html

[8] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[9] http://www.networkworld.com/news/2003/0714boeing.html?page=2

[10] http://www.w3.org/Signature/

[11] http://www.ncipher.com/company/pr_view.php?itemid=218

[12] http://www.broadcom.com/press/release.php?id=665679

[13] http://www.w3.org/TR/xkms/

4. roll-out content protection as needed, using a common cryptographic architecture – to harden data-protection and refine data access.

The reference architecture is designed to be minimalist but must be deployed uniformly if the savings in capital and operating costs are to be optimized. It enables economies of scale based on the general principles of[14]:

> **Demand-driven networking**, which allows network maintenance and management overhead to scale with the business, rather than being high fixed costs, by leveraging high-bandwidth open networks and access on demand technologies;
>
> **Free interconnection** of all types of application, platform and device, reducing the burden and inflexibility of current proprietary integration and increasing the utility and value of the resulting network;
>
> **Universal data interchange** unshackling the costs of proprietary representations and minimizing data transport and conversion costs, using free and open standards, based on XML and associated technologies.

These are the first pragmatic stepping stones that prepare for the nirvana of role-based, fine-grained access to business function and data.

The reference architecture specifically does not consider hardening the operating system environment of PCs and servers, for example to protect the integrity of programs running there, or to reduce the threat from malware. It is assumed that little substantively different can be achieved on today's hardware and operating system platforms, although some improvement is possible by vigorous application of mutual authentication of a tightly controlled set off communicating actors.

There is promise of significant advances on this issue with the roll-out of hardware-protected devices in standard computing equipment, notably the Trusted Platform Module (TPM), which has been defined by the Trusted Computing Group (TCG) standards body[15].

The impact is likely to be in two phases:-

- Firstly TPMs will, at low cost, significantly harden authentication and encryption, for it is widely recognized that hardware-protected cryptographic keys and processes are necessary best practice[16,17].
- Secondly, operating system developments offer the promise of raising the bar for protecting software integrity on standard computers.

This reference architecture has been designed to allow flexible adoption of these developments, so that a corporation can decide when to adopt these approaches.

There is, however, one important general architectural point. The supporting infrastructure must be at least as strong as the protection afforded actors that use the cryptographic keys that underpin credentials and content protection, particularly as they become hardware-protected.

In particular, there is a requirement for strong integrity protection of the identities, their enrollment and associations that are stored in centralized infrastructure such as KPS – and more so for the policy data that

---

[14] Ref 4

[15] https://www.trustedcomputinggroup.org/home

[16] "Key Management Policy and Practice Framework", KPMG Risk and Advisory Services, January 2002, http://www.ncipher.com/insights/km/km_kpmg_intro.html

[17] "Hasten Deployment of Secure Hardware to Maximize PC Security", Gartner, 22 Nov 2004

controls the processes to define and use them. Standard operating system mechanisms must be hardened, for example with specialist hardware security modules, and multiple people may be needed to modify policy data (4-eyed principle).

7 Conclusion

This paper suggests a reference architecture and a phased deployment roadmap that would allow an enterprise to evolve its IT infrastructure towards the Jericho World. It identifies the core building blocks, and standards, which largely are viable today, and suggests that Jericho might foster additional standards in the area of content protection for data in files and relational databases.

It is hoped that Jericho might sponsor a test-bed that implements this reference architecture to demonstrate its virtues to the wider corporate audience.