Report

2011-05-16

The Internet Infrastructure Foundation

Kirei 2011:19

Internet Public Key Infrastructure X.509 (PKIX) usage and management

Contents

1 Internet X.509 Public Key Infrastructure

1.1 X.509 (PKIX) certificate usage and management

X.509 is a ITU standard for a public key infrastructure (PKI) which specifies, among other things, formats for public key certificates, certificate requests, certificate revocation lists and certification path validation algorithm. The X.509 standard was primarily designed to support the X.500 structure. However, today's use cases centre mostly around the Internet.

IETF's Public Key Infrastructure (X.509) working group (PKIX WG) has adapted the standard to the requirements and structure of the Internet. RFC 5280[7] specifies the PKIX Certificate and CRL Profile of the X.509v3 certificate standard.

In the Internet context, PKIX has a number of flaws which affects the security and trust of the system. Most notably, the business model which is service-centric, and normally does not include the relying-party, does not encourage good security practices, but rather lower costs for the service provider. This has lead to a gradual erosion from what originally was an assertion of the legal entity behind an Internet service, to be only an assertion of the control of the domain name. There are also technical implementation flaws, such as the user-agent not requiring successful revocation control and casual overriding of validation failures.

Recent attacks on registration authority functions and the introduction of DNSSEC into the domain name system has put additional focus on these issues, and opened up to different approaches for securing Internet communications.

This report takes an unscientific approach to producing an overview of how X.509 certificates are used in the context of secure Internet communications, and how these certificates are managed. Finally, the report also makes an effort to suggest future developments to better support the new business- and security models which are being developed.

1.1.1 PKI participants

In a Public Key Infrastructure (PKI) the **Subject** is the entity who holds a public key certificate containing the assertion of the Subjects identity.

The **Certification Authority** (CA) is a trusted third party whose main role is to establish a binding between a public key and a Subject (user or service). This is done by signing the Subjects public key (enclosed with information to identify the subject) using the CA's own private key, producing the Subjects public key certificate.

The mechanism that identifies and validates the binding between the public key and the subject is the **Registration Authority** (RA), which may or may not be separate from the CA. The key-user binding is verified, depending on the level of assurance the binding has, by automatic mechanisms or under human supervision.

The **Relying Party** is any entity which validates the Subjects identity using the public key certificate, and relies on the binding of that assertion.

1.1.2 Certificate enrolment procedures

A service provider who wishes to enrol for a public key certificate contacts a Certification Authority which is trusted by the community for which the service provider needs to be authenticated.

The service provider then proves its identity to a certain assurance level defined by that CA's Certification Policy (CP). The assurance level selected may vary depending on the risks associated with the service.

After completing the identification process, the service provider generates a public-private key pair. The public key and a claimed identity of the service is sent to the RA, which validates that the requesting entity is eligible to represent that claimed identity. The RA then forwards the request the CA, which may apply additional information according to the policy, such as where a client can retrieve certificate status information, the identity of the policy under which the CA operates, or for which purposes the certificate may be trusted. The CA then signs the request, producing the clients public key certificate, which is sent back to the requestor (see figure 1.1).

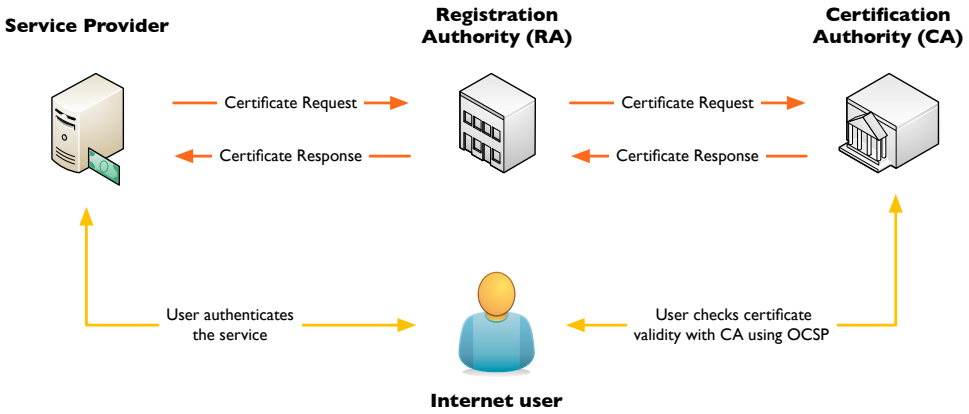

Figure 1.1: Typical application of PKIX certificates on the Internet

By supplying this public key certificate to a contacting client, while also proving possession of the private key using a digital signature scheme, the service provider can authenticate itself to the client and the parties may subsequently exchange encryption keys with each other to establish a secure communications channel.

1.1.3 The Internet use case

In the Internet use case, the service provider is typically running a web service which it wishes to offer securely to the average Internet user. The security goals are most often to prevent others from eavesdropping or modifying the communication, which is achieved by using encryption. For this encryption to be effective, there has to be an authentication mechanism to protect from man-in-the-middle (MITM) attacks. In the Internet web use case, this mechanism is implemented in the user's browser and works by matching the Common Name (CN) attribute of the public key certificate to the requested Fully Qualified Domain Name (FQDN).

Since it is the user's web browser that validates the identity of the web service, it is the web browser which determines what Certification Authorities are to be trusted. This list of trusted third-parties normally comes pre-populated from the browser vendor or with the user's operating system. Hence, the user is normally not in the position of selecting who to trust. Instead, this system works by each of the major browser vendors (currently Microsoft's Internet Explorer, Mozilla Foundation's Firefox, Google's Chrome and Apple's Safari together holds approximately 95% the of market shares) having its own Root CA program, where a CA who wishes to be included has to enrol. To qualify for inclusion in the list, a CA has to adhere to certain principles, such as the principles of the WebTrust framework or equivalent. To prove adherence, the CA has to present a report from

an independent accredited auditor, stating that all controls relevant to the assurance levels offered by the CA is correctly implemented and functional.

The most common assurance level available among the CAs included with the web browsers are the so called **Domain Validation** (DV) level. The common ground requirement for this level of assurance is for the Subject to have proven control over the domain name for which the Subject wishes to enrol a certificate. In some cases, this is verified by just being able to respond to e-mail sent to the postmaster address of that domain (or some other common administration address). Other CAs have a thorough identification process which involves checking of the legal entity and the authority of the representative of that entity. In these cases, the CA may include additional information in the public key certificate, such as the organisation name and location. However, the certificate are validated using the same mechanism as the certificate issued just using the postmaster address, and it would require user-interaction uncommon to the average Internet user to spot the difference.

For this reason, a higher assurance level has been established called **Extended Validation** (EV). A CA who wishes to issue EV certificates under the Root CA program for each browser vendor has to adhere to more strict principles and a comprehensive identification process of the Subject. EV certificates always contains the legal name of the Subject, and it is explicitly displayed in the navigation bar of the web browser.

1.1.4 Implications of compromised RAs or CAs

In the Internet use case described previously, a CA included in the list of trusted third parties can serve the whole Internet market. There are currently over a 100 Root CAs included in the set common to the

major browsers, each of them with numerous intermediate issuing CAs. History brings us that not all of these Certification Authorities can be trusted.

Since any certificate validation chain which ends in one of the pre-installed Root CAs will validate for any service on the Internet, an attacker which are in control of a Certification Authority or any of the affiliated Registration Authorities is able to impersonate or perform a MITM on any service on the Internet.

1.2 Cryptographic protocols for secure Internet communications

Fundamentally, PKIX certificates are used for validating the identity or identities of the communicating parties, and optionally establishing secure keying material for protection of a message or a communications channel.

The PKIX applications can be taxonomically organised according to level 2, 3, 4 and 7 of the OSI model; the link layer, the network layer, the transport layer and the application layer.

1.2.1 Transport layer security

Authentication and establishment of a secure communications channel on top of TCP is probably the most common application of PKIX on the Internet. HTTP as the most prominent of the Internet data communications protocols takes advantage of the Transport Layer Security protocol (TLS, RFC 5247[2]) or the Secure Sockets Layer protocol (SSL) by wrapping the communication in the secure channel established using an URI scheme, and validating authenticity to the pre-configured set of certification authorities in the clients browser (or operating system).

TLS/SSL has the property of authenticating either just the server side, or mutual authentication of both server and client, followed by the establishing of a secure communications channel between the endpoints.

Other commonly used protocols which implements TLS/SSL or uses TLS/SSL as a wrapper is SMTP[17], IMAP[8], XMPP[28] and SIP[24], to mention a few.

1.2.2 Network layer security

Network layer security is primarily used for interconnecting a client to a network or a network to another network, over insecure communication links – a technique generally referred to as Virtual Private Network, or VPN. The most common technology, with wide spread support in networking equipment, is IPsec. IPsec incorporates security directly on top of IP using the Encapsulating Security Payload (ESP[16]) protocol. For authentication of the communicating parties, and negotiation of algorithms and keying material, the Internet Key Exchange protocol (IKE[15]) is commonly used. IKE has support for PKIX certificates.

Due to the limitations imposed by not having endpoint multiplexing (i.e., port numbers), other technologies has evolved to better deal with networking equipment such as NAT devices and firewalls. Common for these technologies is that they tunnel IP over a transport protocol, such as UDP or TCP. For this reason, it is not uncommon to encapsulate IP, or even a link-layer protocol over TCP using TLS/SSL. This technology is often called SSL-VPN, and inherently also often relies upon PKIX certificates for authentication and keying.

1.2.3 Data link layer security

Protection on the data-link layer is most common in wireless network environments, using the 802.11i Robust Security Network (RSN). However, the relatively recent IEEE 802.1AE defines connectionless data confidentiality and integrity for wired networks as well.

Both 802.11i and 802.1AE uses 802.1X for authentication and key management. 802.1X incorporates the Extendable Authentication Protocol (EAP) which in turn has a number of mechanisms based on PKIX certificates for authentication. Most common of those mechanisms is EAP-TLS, EAP-TTLS and PEAP.

Due to the fact that there is no way of mapping a certificate to a unique network identity (e.g., 802.11 Service Set Identifiers, SSID, is an unmanaged name space), one can argue that the security model in this use case is flawed. Just presenting a valid certificate is not enough to authenticate a network. For this reason, user interaction (or pre-configuration) for each network is always needed, or the authentication may more resemble "leap-of-faith" than PKIX validation. Flaws also exist in the implementation of the revocation checking mechanisms. Even though certificate status checking is supported as an integral part of the TLS protocols, it is not strictly required and very few (if any) supplicants enforces it.

1.2.4 Message level protection

The Cryptographic Message Syntax (CMS) is the IETF standard (specified in RFC 5652[13]) for cryptographically protected messages. It can be used to digitally sign, digest, authenticate or encrypt any form of digital data. It is based upon PKCS#7, which is documented in RFC 2315[14].

CMS uses ASN.1 for representing the data structures, and is the

foundation of several other applications which relies upon message level protection. One prominent example is Secure/Multipurpose Internet Mail Extensions (S/MIME) for secure e-mail. Another relatively common application is the signing and encryption of PDF documents, which is based on the CMS's predecessor PKCS#7.

As an alternative method for message level protection, there exists XML-DSig (RFC 3275[1]) which specifies a XML syntax and processing rules for creating and representing digital signatures.

1.2.5 Code signing

The security mechanisms for code signing can be seen as a derivative of message level protection, where the message to be protected/authenticated is executable code.

Executable code may be signed in a number of different ways, depending on what platform or in what environment the executable file is designed to run in. Some of the most common mechanisms includes Microsoft Authenticode, Apple code signing, Java- and JavaScript JAR signing and Mozilla NSS signing.

Typically, the signing process includes bundling and canonisation of the data to be signed and producing either a detached PKCS#7/CMS signature as part of this bundle, or an embedded signature in the file with the executable code.

1.3 Certificate request and enrolment mechanisms

1.3.1 Manual

The most common way of requesting a certificate from a certification authority is manual enrolment using a key generation/certificate signing request (CSR) generation tool, producing a PEM-encoded CSR (PKCS#10) and a private-public key pair. This request is copied and

pasted into a form at the certification authority's enrolment web page, and submitted.

The resulting certificate is then sent back to the client, using the web service, e-mail or some other means, which completes the enrolment.

1.3.2 Using built-in browser functions

In HTML5, the <keygen> tag is defined for generating a private-public key pair in the clients web browser, and post back a certificate request in the SPKAC (SignedPublicKeyAndChallenge) format. The SPKAC request includes proof-of-possession of the private key. It is expected that the user agent stores the generated private key in a key store, and that the server generates and offers back a client certificate to match that private key.

The <keygen> tag is supported in most wide-spread web browsers, with the exception of Microsoft Internet Explorer. For Microsoft Internet Explorer there is an ActiveX control called xenroll (Windows XP and older) or certenroll (Windows Vista and newer) which is able to produce a CMC request or PKCS#10 CSR, but can also selectively enrol the user or the machine. It has significantly more features than the <keygen> tag in this context.

1.3.3 Programatically using certificate management protocols

For the purpose of a more automated approach of enrolling devices and users into a PKI, there has been efforts within the IETF PKIX Working Group to develop certificate management protocols:

CMP – Certificate Management Protocol specified by the IETF in RFC 4210[3]. It is based on CMS and uses the CRMF (RFC

4211[30]) format for certificate signing requests. It seems to have gained very little traction and market support, and is primarily used in closed smart-card enrolment environments. CMP carries heavy complexity, which may explain the low popularity. CMP is a message-level protocol, and its transport is not defined.

CMC – Certificate Management over CMS specified by the IETF in RFC 5272[31], and like CMP, uses CMS and the CRMF format for certificate signing requests. It is used internally in Microsoft Windows environments. CMC transport mechanisms are defined in RFC 5273[32], using HTTP, file, e-mail and TCP.

SCEP – Simple Certificate Enrollment Protocol is the de-facto standard for enrolling devices and users in a simple and straight-forward manner . It uses HTTP as transport, and PKCS#7 messages. Paradoxically, after 10 years in the IETF PKIX WG, it is still a draft. There has been recent discussions of finalising it, and move it to a RFC.

SCEP is supported by some of the largest certification authorities on the market, as well as a broad base of manufacturers of network devices. There are open source software libraries available for custom integrations. CA software solutions, such as the Microsoft CA, OpenCA and Red Hat CA system, among others, all support SCEP enrolment.

Simplicity is most likely what founded the popularity of SCEP.

Some within the IETF PKIX WG argue in favor for a simple profiling of CMC, which offers functionality similar to SCEP, but based on more modern URI scheme and CMS format messages.

1.4 Prevalent server applications of PKIX certificates

The most popular server software applications which incorporates Transport Layer Security (TLS/SSL) on the Internet can be organised into three categories:

OpenSSL environments, which commonly includes the Apache web server, Postfix Mail Transfer Agent (MTA), Dovecot IMAP/POP e-mail server, Qmail MTA, Nginx web server and Courier MTA/IMAP/POP e-mail server (among several others).

Even though any tool could be used to enrol a certificate for these server applications, the administrators' natural choice is often OpenSSL. OpenSSL provides cryptographic libraries and a command-line tool which enables the administrator to manage certificates with large flexibility. OpenSSL is often criticised to be anything but user-friendly for the user, and a helter-skelter for the developer. The key store most often consists of a plain file (in DER or PEM format), protected with the discretionary or mandatory file access control of the operating system. CA certificates are typically configured per server daemon, or imported from a catalog with trusted entities provided by the operating system vendor/distributor.

Microsoft Windows which carries the Internet Information Server (IIS) web server and the Exchange MTA and mail access server has built-in functionality for enrolling a certificate.

Each of the application systems on the Microsoft Windows platforms generally provides a wizard for generating certificate requests and installing the signed certificate into the proper key store integrated with the operating system. The key stores on Windows is a database protected with the operating systems

access control system, and includes a pre-installed set of trusted third party CA certificates.

Java is represented by a large base of web application server frameworks, including JBoss, WebLogic, WebSphere.

Java provides a key store file, and a management command-line tool called keytool. Administrators can generate keys, certificate requests, and install end entity certificates and CA certificates into the key store. The list may come empty or pre-populated with trusted third parties CA certificates, depending on the vendor/distributor.

Mac OS X has a certificate assistant built into the keystore management application, which is capable of generating keys, self-signed certificates and certificate requests, among other things. The Mac OS X key store comes pre-populated with trusted third parties CA certificates.

So far the number of server applications which uses the OS X key store seems limited, and the functionality is mostly used for administration of OS X environments, rather than Internet services.

1.5 Future development

1.5.1 Mitigating risks associated with compromised CAs

In the Internet use case described in section 1.1.3, and with the associated vulnerabilities elaborated on in section 1.1.4, it is desirable for service providers to be able to preclude or limit the acceptable validation paths. This can be achieved by using secure DNS for publishing one or more root certificates where the validation may end, otherwise the certificate should be rejected by the client.

This limits the vulnerability surface to only the CA or CAs where the service provider is in fact a customer, in practice reducing the root CAs from over a 100 to one, or just a few.

The effect for the service provider can be even better by selecting a CA with good security practices, as this CA is less likely to experience a security breach.

This simple extension to the certificate validation mechanisms in the browser is expected to considerably reduce the vulnerabilities of the current PKIX model, and also to some extent affect the business model as better CA security practices provides a business case and may be rewarded by attracting security-aware customers.

1.5.2 Using Secure DNS to Associate Certificates with Domain Names For TLS (DANE)

Since the assertion embodied in the PKIX Domain Validation certification boils down to control over the domain name, and with the advent of DNSSEC, validation may be directly tied into DNS instead of using a Certification Authority. This has a number of advantages; as opposed to using the long-lived assertions of a CA, the assertion published in DNS would be momentary, mitigating the certificate status checking vulnerabilities and increasing the service's responsiveness. Tying the validation into DNS can also preclude validation paths to the clients pre-installed list of CA's.

This move from using a local list of supposedly trusted third parties, into validating the service's certificate directly in the DNS, calls for a number of changes in how certificates are enrolled and validated. It is expected that this change will happen gradually, as service providers start publishing their services' certificates in DNS and clients gain support for the new technology.

These mechanisms are generic for the transport layer, and can be

used to secure any protocol based on TLS, such as SMTP and SIP.

To facilitate generation of keying material and creation of the proper resource records, a signed Java applet which can be run in the system administrators' browser might possibly lower the knowledge threshold for using DANE. This Java applet could generate a self-signed certificate with the proper attributes and extensions for the service in question. Apart from outputting the keying material and the certificate, it could also assist in creating the proper resource records for publication in the DNS.

1.5.3 Using Secure DNS to Associate Certificates with e-mail addresses

Secure DNS can also be used to bind public keys to e-mail addresses, hence enabling secure email based on S/MIME. As there are wide support for S/MIME in todays Mail User Agents (MUAs), the required changes would be limited to the mechanisms for looking up and validating certificates, and extend this into using DNS as well.

Once there is a standard for how S/MIME using DNS is done, a web page could assist end-users with creating a digital certificate for their own e-mail address, using nothing but the user's browser. The web server would include a dummy CA which would sign any certificate request, regardless of the claimed identity, since this signature would not be in the trust chain and hence never validated. The web service could output not only the user's certificate, but also the DNS resource record to be published in the user's e-mail providers zone.

In the far future, using this technology may also enable for the end-to-end protection mechanisms of SIP signalling (and subsequently for reliable confidentiality of voice communications) which is also based on S/MIME. The obvious obstacles along this

path is the enabling of this functionality in the SIP UA's (i.e., the VoIP phones), and other not yet invented mechanisms for key management and DNS integration into these devices.

1.5.4 Use of client certificates in cloud services

Using the previously described mechanisms for securing e-mail would still be a problem for the extensively popular web-mail services. For accessing a web-mail service, the user uses the web browser instead of a full-fledged MUA.

However, there are no generic or standardised support of performing client-side cryptographic operations in the browsers environment, based on the user's key store. More specifically, the de-facto standard for client-side scripting language for web pages is JavaScript, which lacks the capability of interacting with any cryptographic functions in the browser, even though the browser engine itself may have this support. Maybe most fundamentally, a strong PRNG (pseudo-random number generator) can not be implemented in JavaScript due to the nature of the environment it is executed in. Also, there is no way of requesting cryptographic operations from a browsers key store.

An interface similar to PKCS#11 exposed to the JavaScript engine would enable this kind of functionality, which could be used for handling encryption/decryption of server-side data, something which could dramatically improve the security model of cloud services, including support for web mail S/MIME and encrypted on-line storage where the service provider does not hold the encryption keys.

Such standardised functionality would also aid in providing better tools for enrolling users and services for certificates.

A Definitions

801.1AE is the IEEE MAC Security standard (also known as MACsec) which defines connectionless data confidentiality and integrity for media access independent protocols.

802.11i specifies security mechanisms for wireless networks.

802.1X is an IEEE Standard for port-based Network Access Control (PNAC) and provides an authentication mechanism to devices wishing to attach to a LAN or WLAN.

ASN.1 Abstract Syntax Notation One is a standard and flexible notation that describes data structures for representing, encoding, transmitting, and decoding data. It provides a set of formal rules for describing the structure of objects that are independent of machine-specific encoding techniques and is a precise, formal notation that removes ambiguities.

CA Certification Authority (CA) is an entity that issues digital certificates in a PKI.

CP Certificate Policy (CP) is a document which aims to state what the different actors of a public key infrastructure (PKI) are, their roles and their responsibilities. This document is published in the PKI perimeter

CMC Certificate Management over CMS (CMC) is an internet standard by the IETF for management of PKI function.

Additionally, it is defining transport mechanisms for the Cryptographic Message Syntax (CMS), and uses the CRMF format for certificate signing requests. Its specification is RFC 5272[31] and RFC 5273[32].

CMP Certificate Management Protocol (CMP) is an Internet protocol used for obtaining X.509 digital certificates in a public key infrastructure (PKI), and uses the CRMF format for certificate signing requests. Its specification is RFC 4210[3].

CMS Cryptographic Message Syntax (CMS) is the IETF's standard for cryptographically protected messages defined in RFC 5652[13]. It can be used to digitally sign, digest, authenticate or encrypt any form of digital data. CMS is based on the syntax of PKCS#7

CRMF Certificate Request Message Format (CRMF) is a format of messages sent to a certification authority to request certification of a public key, defined in RFC 4211[30]. See CSR.

CSR Certificate Signing Request (CSR) is a message sent from an applicant to a certificate authority in order to apply for a digital identity certificate.

DNSSEC The Domain Name System Security Extensions (DNSSEC) is a suite of Internet Engineering Task Force (IETF) specifications for securing certain kinds of information provided by the Domain Name System (DNS) as used on Internet Protocol (IP) networks and is specified in RFC 4033[4], RFC 4034[6], and RFC 4035[5]. It is a set of extensions to DNS which provide to DNS clients (resolvers) origin authentication of DNS data, authenticated denial of existence, and data integrity, but not availability or confidentiality.

EAP-PEAP Protected Extensible Authentication Protocol (PEAP) is a chaining mechanism which wrappes the EAP protocol within TLS, providing security for the inner EAP session, similarily to EAP-TTLS.

EAP-TLS EAP-Transport Layer Security (EAP-TLS) is an EAP method which uses PKI to mutually authenticate an authentication server and a user. It is defined in RFC 5216[33].

EAP-TTLS EAP-Tunneled Transport Layer Security (EAP-TTLS) is an EAP method that encapsulates a TLS session where the server is authenticated to the client using standard TLS procedures, which subsequently is used for securing the information exchange where the client is authenticated to the server using an arbitrary authentication mechanism encapsulated within the secure tunnel. EAP-TTLS is defined in RFC 5281[11]

EAP Extensible Authentication Protocol (EAP), is an authentication framework frequently used in wireless networks and Point-to-Point connections.

ESP Encapsulating Security Payload (ESP) is a member of the IPsec protocol suite and specified in RFC 2406[16]. It provides origin authenticity, integrity, and confidentiality protection of packets.

Firewall (in the context of computing) is a device or set of devices designed to permit or deny network transmissions based upon a set of rules and is frequently used to protect networks from unauthorised access while permitting legitimate communications to pass.

FQDN Fully Qualified Domain Name (FQDN) is a domain name that specifies its exact location in the tree hierarchy of the Domain Name System (DNS). It specifies all domain levels, including

the top-level domain and the root domain. A fully qualified domain name is distinguished by its unambiguity; it can only be interpreted one way.

HTML HyperText Markup Language, is the predominant markup language for web pages. HTML is the basic building-blocks of webpages.

HTTP Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web and was originally specified in RFC 2616[10].

IETF The Internet Engineering Task Force (IETF) develops and promotes Internet standards.

IKE Internet Key Exchange (IKE or IKEv2) defined in RFC 2409[12] and RFC 4306[15] is a protocol used to set up a security association (SA) in the IPsec protocol suite.

IMAP Internet message access protocol (commonly known as IMAP) is one of the two most prevalent Internet standard protocols for e-mail retrieval. IMAP version 4 revision 1 (IMAP4rev1), is defined in RFC 3501[8].

IPsec Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session.

IP Internet Protocol (IP) is the principal communications protocol used for relaying datagrams (packets) across an internetwork

using the Internet Protocol Suite. IP is defined in RFC 791[21] and is responsible for routing packets across network boundaries, it is the primary protocol that establishes the Internet.

JAR Java ARchive is a file format which aggregates many files into one.

MITM Man-In-The-Middle attack (MITM) or sometimes Janus attack, is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. The attacker must be able to intercept all messages going between the two victims and inject new ones.

MTA Mail Transfer Agent (MTA) or mail relay is software that transfers electronic mail messages from one computer to another using a client–server application architecture. An MTA implements both the client (sending) and server (receiving) portions of the Simple Mail Transfer Protocol.

NAT Network Address Translation (NAT) is the process of modifying IP address information in IP packet headers while in transit across a traffic routing device.

NSS Network Security Services (NSS) comprises a set of libraries designed to support cross-platform development of security-enabled client and server applications.

OSI model Open Systems Interconnection model is a way of subdividing a communications system into smaller parts called layers. Similar communication functions are grouped into

logical layers. A layer provides services to its upper layer while receiving services from the layer below.

PDF Portable Document Format (PDF) is an open standard for document exchange.

PEM Privacy Enhanced Mail (PEM), is a 1993 IETF proposal for securing email using public key cryptography. Although PEM became an IETF proposed standard it was never widely deployed or used.

PKCS#10 Public Key Cryptography Standard number 10 is a format of messages sent to a certification authority to request certification of a public key. See CSR.

PKCS#11 Public Key Cryptography Standard number 11 (also known as cryptoki) is an API defining a generic interface to cryptographic tokens.

PKCS#7 Public Key Cryptography Standard number 7 specifies is a standard for cryptographically protecting messages.

PKIX The Public Key Infrastructure (X.509) working group (PKIX) is a working group of the Internet Engineering Task Force dedicated to creating standard documentation on issues related to public key infrastructure based on X.509 certificates.

PKI Public Key Infrastructure is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

POP Post Office Protocol (POP) is an Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection. Its current specification is RFC 1939[19].

PRNG Pseudorandom Number Generator (PRNG), also known as a deterministic random bit generator (DRBG), is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state.

RSN See 802.11i.

SA A Security Association (SA) is the establishment of shared security attributes between two network entities to support secure communication.

SCEP Simple Certificate Enrollment Protocol is a protocol for issuing and revocation of digital certificates.

SIP Session Initiation Protocol (SIP) is an IETF-defined signaling protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol (IP). Its latest specification is RFC 3261[24].The protocol can be used for creating, modifying and terminating two-party (unicast) or multiparty (multicast) sessions consisting of one or several media streams.

SMTP Simple Mail Transfer Protocol (SMTP) is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks. SMTP was first defined by RFC 821[23] (1982, eventually declared STD 10), and last updated by RFC 5321[17].

SPKAC Signed Public Key and Challenge, is a format of messages sent to a certification authority to request certification of a public key. See CSR.

SSID Service Set Identifier (SSID) is a name that identifies a particular 802.11 wireless LAN.

SSL See TLS.

TCP Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite and is originally specified in RFC 793[22]. TCP is one of the two original components of the suite, complementing the Internet Protocol (IP), and therefore the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer.

TLS Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communications security over the Internet. Version 1.2 of the TLS protocol is specified in TLS 1.2 was defined in RFC 5246[9]. TLS and SSL encrypt the segments of network connections above the Transport Layer, using symmetric cryptography for privacy and a keyed message authentication code for message reliability.

UDP User Datagram Protocol (UDP) defined in RFC 768[20]is one of the core members of the Internet Protocol Suite, the set of network protocols used for the Internet. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without requiring prior communications to set up special transmission channels or data paths

URI Uniform Resource Identifier (URI) (further described in RFC 3305[18] is a string of characters used to identify a name or a resource on the Internet. Such identification enables interaction with representations of the resource over a network (typically the World Wide Web) using specific protocols. Schemes

specifying a concrete syntax and associated protocols define each URI.

VPN Virtual Private Metwork (VPN) is a secure way of connecting to a private Local Area Network at a remote location, using the Internet or any insecure public network to transport the network data packets privately, using encryption. The VPN uses authentication to deny access to unauthorised users, and encryption to prevent unauthorised users from reading the private network packets.

X.509 is an ITU-T standard for a public key infrastructure (PKI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

XML-DSig XML Signature (also called XMLDsig, XML-DSig, XML-Sig) defines an XML syntax for digital signatures and is defined in the W3C recommendation XML Signature Syntax and Processing. Functionally, it has much in common with PKCS#7 but is more extensible and geared towards signing XML documents.

XML Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form.

XMPP Extensible Messaging and Presence Protocol (XMPP) is an open-standard communications protocol for message-oriented middleware based on XML (Extensible Markup Language) and is defined by the five specifications RFC 3922[26], RFC 3923[25], RFC 6120[28] and RFC 6121[29] and RFC 6122[27]. The protocol was originally named Jabber, and was developed by the Jabber open-source community in 1999 for, originally, near-real-time,

extensible instant messaging (IM), presence information, and contact list maintenance. Designed to be extensible, the protocol today also finds application in VoIP and file transfer signaling.

Source of most definitions; Wikipedia: The Free Encyclopedia.

B References

[1] (Extensible Markup Language) XML-Signature Syntax and Processing, D. Eastlake 3rd, J. Reagle, and D. Solo. RFC 3275 (Draft Standard), March 2002. URL: http://www.ietf.org/rfc/rfc3275.txt

[2] Extensible Authentication Protocol (EAP) Key Management Framework, B. Aboba, D. Simon, and P. Eronen. RFC 5247 (Proposed Standard), August 2008. URL: http://www.ietf.org/rfc/rfc5247.txt

[3] Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP), C. Adams, S. Farrell, T. Kause, and T. Mononen. RFC 4210 (Proposed Standard), September 2005. URL: http://www.ietf.org/rfc/rfc4210.txt

[4] DNS Security Introduction and Requirements, R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. RFC 4033 (Proposed Standard), March 2005. Updated by RFC 6014. URL: http://www.ietf.org/rfc/rfc4033.txt

[5] Protocol Modifications for the DNS Security Extensions, R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. RFC 4035 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014. URL: http://www.ietf.org/rfc/rfc4035.txt

[6] Resource Records for the DNS Security Extensions, R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. RFC 4034 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014. URL: http://www.ietf.org/rfc/rfc4034.txt

[7] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. RFC 5280 (Proposed Standard), May 2008. URL: http://www.ietf.org/rfc/rfc5280.txt

[8] INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1, M. Crispin. RFC 3501 (Proposed Standard), March 2003. Updated by RFCs 4466, 4469, 4551, 5032, 5182, 5738, 6186. URL: http://www.ietf.org/rfc/rfc3501.txt

[9] The Transport Layer Security (TLS) Protocol Version 1.2, T. Dierks and E. Rescorla. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176. URL: http://www.ietf.org/rfc/rfc5246.txt

[10] Hypertext Transfer Protocol – HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785. URL: http://www.ietf.org/rfc/rfc2616.txt

[11] Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0), P. Funk and S. Blake-Wilson. RFC 5281 (Informational), August 2008. URL: http://www.ietf.org/rfc/rfc5281.txt

[12] The Internet Key Exchange (IKE), D. Harkins and D. Carrel. RFC 2409 (Proposed Standard), November

1998. Obsoleted by RFC 4306, updated by RFC 4109. URL: http://www.ietf.org/rfc/rfc2409.txt

[13] Cryptographic Message Syntax (CMS), R. Housley. RFC 5652 (Standard), September 2009. URL: http://www.ietf.org/rfc/rfc5652.txt

[14] PKCS #7: Cryptographic Message Syntax Version 1.5, B. Kaliski. RFC 2315 (Informational), March 1998. URL: http://www.ietf.org/rfc/rfc2315.txt

[15] Internet Key Exchange (IKEv2) Protocol, C. Kaufman. RFC 4306 (Proposed Standard), December 2005. Obsoleted by RFC 5996, updated by RFC 5282. URL: http://www.ietf.org/rfc/rfc4306.txt

[16] IP Encapsulating Security Payload (ESP), S. Kent and R. Atkinson. RFC 2406 (Proposed Standard), November 1998. Obsoleted by RFCs 4303, 4305. URL: http://www.ietf.org/rfc/rfc2406.txt

[17] Simple Mail Transfer Protocol, J. Klensin. RFC 5321 (Draft Standard), October 2008. URL: http://www.ietf.org/rfc/rfc5321.txt

[18] Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations, M. Mealling and R. Denenberg. RFC 3305 (Informational), August 2002. URL: http://www.ietf.org/rfc/rfc3305.txt

[19] Post Office Protocol - Version 3, J. Myers and M. Rose. RFC 1939 (Standard), May 1996. Updated by RFCs 1957, 2449, 6186. URL: http://www.ietf.org/rfc/rfc1939.txt

[20] User Datagram Protocol, J. Postel. RFC 768 (Standard), August 1980. URL: http://www.ietf.org/rfc/rfc768.txt

[21] Internet Protocol, J. Postel. RFC 791 (Standard), September 1981. Updated by RFC 1349. URL: http://www.ietf.org/rfc/rfc791.txt

[22] Transmission Control Protocol, J. Postel. RFC 793 (Standard), September 1981. Updated by RFCs 1122, 3168, 6093. URL: http://www.ietf.org/rfc/rfc793.txt

[23] Simple Mail Transfer Protocol, J. Postel. RFC 821 (Standard), August 1982. Obsoleted by RFC 2821. URL: http://www.ietf.org/rfc/rfc821.txt

[24] SIP: Session Initiation Protocol, J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141. URL: http://www.ietf.org/rfc/rfc3261.txt

[25] End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP), P. Saint-Andre. RFC 3923 (Proposed Standard), October 2004. URL: http://www.ietf.org/rfc/rfc3923.txt

[26] Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM), P. Saint-Andre. RFC 3922 (Proposed Standard), October 2004. URL: http://www.ietf.org/rfc/rfc3922.txt

[27] Extensible Messaging and Presence Protocol (XMPP): Address Format, P. Saint-Andre. RFC 6122 (Proposed Standard), March 2011. URL: http://www.ietf.org/rfc/rfc6122.txt

[28] Extensible Messaging and Presence Protocol (XMPP): Core, P. Saint-Andre. RFC 6120 (Proposed Standard), March 2011. URL: http://www.ietf.org/rfc/rfc6120.txt

[29] Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, P. Saint-Andre. RFC 6121 (Proposed Standard), March 2011. URL: http://www.ietf.org/rfc/rfc6121.txt

[30] Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF), J. Schaad. RFC 4211 (Proposed Standard), September 2005. URL: http://www.ietf.org/rfc/rfc4211.txt

[31] Certificate Management over CMS (CMC), J. Schaad and M. Myers. RFC 5272 (Proposed Standard), June 2008. URL: http://www.ietf.org/rfc/rfc5272.txt

[32] Certificate Management over CMS (CMC): Transport Protocols, J. Schaad and M. Myers. RFC 5273 (Proposed Standard), June 2008. URL: http://www.ietf.org/rfc/rfc5273.txt

[33] The EAP-TLS Authentication Protocol, D. Simon, B. Aboba, and R. Hurst. RFC 5216 (Proposed Standard), March 2008. URL: http://www.ietf.org/rfc/rfc5216.txt