Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.\[1\] In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a certificate authority (CA). The user identity must be unique within each CA domain. The binding is established through the registration and issuance process, which, depending on the level of assurance the binding has, may be carried out by software at a CA, or under human supervision. The PKI role that assures this binding is called the Registration Authority (RA). For each user, the user identity, the public key, their binding, validity conditions and other attributes are made unforgeable in public key certificates issued by the CA.

The term trusted third party (TTP) may also be used for certificate authority (CA). The term PKI is sometimes erroneously used to denote public key algorithms, which do not require the use of a CA.

**Alternatives**

Broadly speaking, there are three approaches to getting this trust: Certificate Authorities (CAs), Web of Trust (WoT), and Simple public key infrastructure (SPKI).

**Certificate Authorities**

The primary role of the CA is to publish the key bound to a given user. This is done using the CA's own key, so that trust in the user key relies on one's trust in the validity of the CA's key. The mechanism that binds keys to users is called the Registration Authority (RA), which may or may not be separate from the CA. The key-user binding is established, depending on the level of assurance the binding has, by software or under human supervision.

The term trusted third party (TTP) may also be used for certificate authority (CA). Moreover, PKI is itself often used as a synonym for a CA implementation.

**Temporary Certificates & Single Sign-On**

This approach involves a server that acts as an online certificate authority within a single sign-on system. A single sign-on server will issue digital certificates into the client system, but never stores them. Users can execute programs, etc. with the temporary certificate. It is common to find this solution variety with x.509-based certificates.\[2\]

**Web of Trust**

An alternative approach to the problem of public authentication of public key information is the web of trust scheme, which uses self-signed certificates and third party attestations of those certificates. The singular term Web of Trust does not imply the existence of a single web of trust, or common point of trust, but rather one of any number of potentially disjoint "webs of trust". Examples of implementations of this approach are PGP (Pretty Good Privacy)
and GnuPG (an implementation of OpenPGP, the standardized specification of PGP). Because PGP and implementations allow the use of e-mail digital signatures for self-publication of public key information, it is relatively easy to implement one's own Web of Trust. One of the benefits of the Web of Trust, such as in PGP, is that it can interoperate with a PKI CA fully-trusted by all parties in a domain (such as an internal CA in a company) that is willing to guarantee certificates, as a trusted introducer. Only if the "web of trust" is completely trusted, and because of the nature of a web of trust, trusting one certificate is granting trust to all the certificates in that web. A PKI is only as valuable as the standards and practices that control the issuance of certificates and including PGP or a personally instituted web of trust could significantly degrade the trustability of that enterprise's or domain's implementation of PKI.[3]

The web of trust concept was first put forth by PGP creator Phil Zimmermann in 1992 in the manual for PGP version 2.0:

As time goes on, you will accumulate keys from other people that you may want to designate as trusted introducers. Everyone else will each choose their own trusted introducers. And everyone will gradually accumulate and distribute with their key a collection of certifying signatures from other people, with the expectation that anyone receiving it will trust at least one or two of the signatures. This will cause the emergence of a decentralized fault-tolerant web of confidence for all public keys.

**Simple public key infrastructure**

Another alternative, which however does not deal with public authentication of public key information, is the simple public key infrastructure (SPKI) that grew out of three independent efforts to overcome the complexities of X.509 and PGP's web of trust. SPKI does not associate users with persons, since the key is what is trusted, rather than the person. SPKI does not use any notion of trust, as the verifier is also the issuer. This is called an "authorization loop" in SPKI terminology, where authorization is integral to its design.

**History**

The concepts and use of Public Key Infrastructure were discovered by British scientists in GCHQ in 1969 with Ellis. After the re-discovery and commercial use of PKI by Rivest, Shamir, Diffie and others, the British government considered releasing the records of GCHQ's successes in this field. However, the untimely publication of Spycatcher meant that the government once again issued a gag order and full details of GCHQ achievement were never revealed. The public disclosure of both secure key exchange and asymmetric key algorithms in 1976 by Diffie, Hellman, Rivest, Shamir, and Adleman changed secure communications entirely. With the further development of high speed digital electronic communications (the Internet and its predecessors), a need became evident for ways in which users could securely communicate with each other, and as a further consequence of that, for ways in which users could be sure with whom they were actually interacting.

Assorted cryptographic protocols were invented and analyzed within which the new cryptographic primitives could be effectively used. With the invention of the World Wide Web and its rapid spread, the need for authentication and secure communication became still more acute. Commercial reasons alone (e.g., e-commerce, on-line access to proprietary databases from Web browsers, etc.) were sufficient. Taher Elgamal and others at Netscape developed the SSL protocol ('https' in Web URLs); it included key establishment, server authentication (prior to v3, one-way only), and so on. A PKI structure was thus created for Web users/sites wishing secure communications.

Vendors and entrepreneurs saw the possibility of a large market, started companies (or new projects at existing companies), and began to agitate for legal recognition and protection from liability. An American Bar Association technology project published an extensive analysis of some of the foreseeable legal aspects of PKI operations (see ABA digital signature guidelines), and shortly thereafter, several US states (Utah being the first in 1995) and other jurisdictions throughout the world, began to enact laws and adopt regulations. Consumer groups and others raised
questions of privacy, access, and liability considerations which were more taken into consideration in some jurisdictions than in others.

The enacted laws and regulations differed, there were technical and operational problems in converting PKI schemes into successful commercial operation, and progress has been far slower than pioneers had imagined it would be.

By the first few years of the 21st century, it had become clear that the underlying cryptographic engineering was not easy to deploy correctly, that operating procedures (manual or automatic) were not easy to correctly design (nor even if so designed, to execute perfectly, which the engineering required), and that such standards as existed were in some respects inadequate to the purposes to which they were being put.

PKI vendors have found a market, but it is not quite the market envisioned in the mid-90s, and it has grown both more slowly and in somewhat different ways than were anticipated. PKIs have not solved some of the problems they were expected to, and several major vendors have gone out of business or been acquired by others. PKI has had the most success in government implementations; the largest PKI implementation to date is the Defense Information Systems Agency (DISA) PKI infrastructure for the Common Access Cards program.

Security issues

see X.509

Usage examples

PKIs of one type or another, and from any of several vendors, have many uses, including providing public keys and bindings to user identities which are used for:

- Encryption and/or sender authentication of e-mail messages (e.g., using OpenPGP or S/MIME).
- Encryption and/or authentication of documents (e.g., the XML Signature [5] or XML Encryption [6] standards if documents are encoded as XML).
- Authentication of users to applications (e.g., smart card logon, client authentication with SSL). There's experimental usage for digitally-signed HTTP authentication in the Enigform and mod_openpgp projects.
- Bootstrapping secure communication protocols, such as Internet key exchange (IKE) and SSL. In both of these, initial set-up of a secure channel (a "security association") uses asymmetric key (a.k.a. public key) methods, whereas actual communication uses faster symmetric key (a.k.a. secret key) methods.
- Mobile signatures[7] are electronic signatures that are created using a mobile device and rely on signature or certification services in a location independent telecommunication environment.

Terminology

- CA: Certificate authority
- TTP: Trusted third party

References

External links

- Public Key Infrastructure Overview (http://www.sun.com/blueprints/0801/publickey.pdf) by Joel Weise - SunPSSM Global Security Practice
- PKI tutorial (http://www.cs.auckland.ac.nz/~pgut001/pubs/pktutorial.pdf) by Peter Gutmann
- PKI Tutorial using Fingerpuppets (http://www.carillon.ca/tutorials.php)
- PKIX workgroup (http://www.ietf.org/html.charters/pkix-charter.html)
- NIST PKI Program (http://csrc.nist.gov/pki/) — in which the National Institute of Standards and Technology (NIST) is attempting to develop a public key infrastructure
  - Response to Ten Risks (http://homepage.mac.com/aramperez/responsetenrisks.html) by A. Perez
  - Seven and a Half Non-risks of PKI (http://www.apache-ssl.org/7.5things.txt) by B. Laurie
- The Inevitable Collapse of the Certificate Model (http://www.hbarel.com/blog/?itemid=36) by Hagai Bar-El

Article Sources and Contributors


Image Sources, Licenses and Contributors


License

Creative Commons: Attribution-Share Alike 3.0 Unported http://creativecommons.org/licenses/by-sa/3.0/