Virtual Private Networks Based Trusted Third Parties Services

Alex Galis, John Lam, Walter Eaves
{a.galis,jolam}@ee.ucl.ac.uk
Department of Electrical & Electronic Engineering
University College London

Abstract: *This paper presents the on-going development effort in the IST project called HARP (Harmonisation for the security of web technologies and applications). It begins by introducing trusted third parties services and it concludes with some open issues and challenges to be addressed in the project.*

1. Trusted Third Party Services

There is a need to facilitate the growing importance and development of electronic commerce, and the European information infrastructure by the introduction of suitable measures to safeguard the integrity and confidentiality of electronic information. The provision of TTPs to satisfy this user need, and the requirement to be compliant with national legislation, is of major importance to establish the right level of user assistance. A TTP has been defined by ISO/IEC as a security authority or its agent trusted by users with respect to security-related activities, e.g. to support the use of digital signatures and confidentiality services.
The TTP solutions are developed in the HARP project [1] by designing VPN security services, tools and mechanisms so TTPs can cope with the diversity of all Web components and provide a harmonised security solution for Web applications involving Web technologies.
The most important TTP-service is the PKI. In the core of any PKI are the *asymmetric encryption and digital signature* techniques. In order for these techniques to be effectively secure, two fundamental assumptions must be true: the private key has to be *unique* and *secret* and other entities must be able to establish trust on the real owner of the key. In order to deal with these issues, *public certificates* have been introduced, as per the X.509 standard.
The certificate is itself signed by a TTP who has to be trusted by every PKI member. TTPs who issue certificates act as Certification Authorities (CAs). The agreements, procedures and policies followed by CAs enable PKI members to use the signature of CAs for establishing trust on the identity of other PKI users and components. Among other duties, CAs guarantee the key uniqueness of both users and components, maintain a database of certificates, *revoke* certificates, maintain a list of revoked certificates, and arrange for the public availability and distribution of certificates.

The use of special-purpose PKI components categorised as either *servers* or *user agents are essential*. The purpose of servers is to store, handle, manage, and provide information of various types. Information can either be made available to all PKI members, or be restricted within a specific region or organisation.
The server components include:

- Certification servers, whose WPs are to accept certification requests, process them, issue certificates, manage and maintain certificates, revoke certificates, maintain one or more Certification Revocation Lists (CRLs), and make certificates and the CRL available to other components.
- Directory servers, that store parts of the X.500 DAP in the form of generalised objects, including documents, addressing information, executable code, certificates and CRLs. DAP requires an X.500 directory, which in many cases is not the most suitable directory solution.

- Web servers, which, in addition to offering generic hypermedia information, provide a versatile and platform-neutral interface for interaction between a user and an arbitrary application over the Internet, by means of the HTTP protocol.

2. The HARP Project and Motivation

TTP services can be considered as value-added communication services available to users that need to enhance the trust in the services used. By signing up to a licensed TTP, the user will be able to communicate securely with every user of every TTP with whom his TTP has an agreement. Therefore, TTPs should be able to offer value with regard to integrity and confidentiality of the electronic information being carried by these communications. The role of TTPs includes providing assurance that:

- messages and transactions are being transferred to the right recipient at the right location,
- messages are received in a timely, secure and accurate manner from the claimed originator/sender,
- for any business dispute that arises, there are appropriate mechanisms for establishing and presenting evidence of what happened.

Users will require TTP services to be available when they need them within the terms of the agreed service contract [2]. TTPs can be categorised according to their communication relationship with the users they serve.

An **off-line** TTP does not interact with the user entities during the process of a given security service. Instead the interaction to provide, or register security-related information is carried out off-line as a separate interaction.

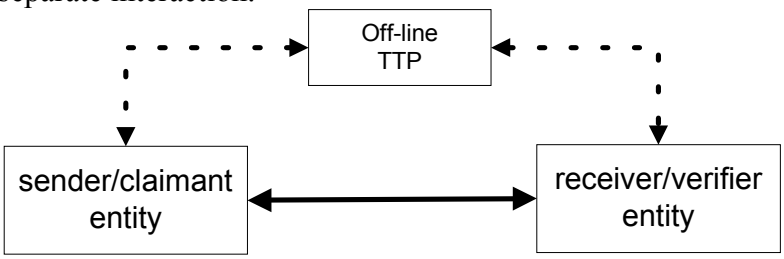

Figure 1: Off-Line TTPs

An **on-line** TTP is requested by one or both entities in real-time to provide, or register, security-related information. Such a TTP is not in the communications path between the two entities.

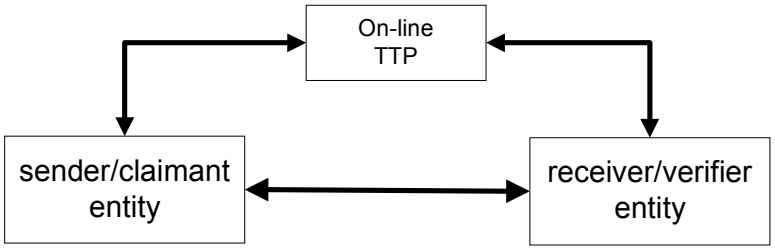

Figure 2: On-Line TTPs

An **in-line** TTP is positioned in the communication path between the entities. Such an arrangement allows the TTP to offer a wide range of security services directly to users. Since the TTP interrupts the communication path, different security domains can exist on either side of it.

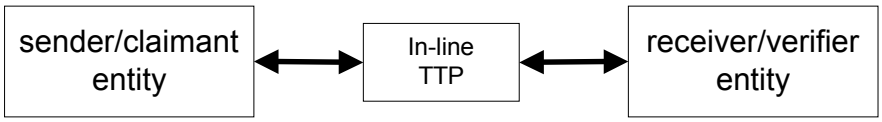

Figure 3: In-Line TTPs

Among, the most widely used software protocols in various security systems are:

- *Domain Name Server Security* (**DNSSEC**). This is a protocol for secure distributed name services. A *Domain Name Server* (**DNS**) transaction is defined by a query/response pair. In a secure computing paradigm, two certificates need to be exchanged to prove each server to each client. It is clear that in recursively traversing a graph of keys, the same model that is underlying the CAs (or TTPs).
- **GSSAPI** (*Generic Security Services API*) provides a generic authentication, key exchange, and encryption interface to different systems and authentication methods.
- **SSL** (*Secure Socket Layer*) is one of the two protocols for secure Web connections (the other is SHTTP). Further information can be found from http://home.mcom.com/info/security-doc.html. IETF has instead developed the TLS specification, which is very similar to SSL. Both SSL and TLS and many other protocols are based on the Diffie-Hellman protocol for key exchange.
- **SHTTP** (*Secure Hypertext Transfer Protocol*) is a complimentary protocol for providing more security for WEB transactions. There is an Internet Draft of the SHTTP protocol.
- **E-Mail security and related services.** These include the S/MIME (Secure-MIME) protocol (http://Web.rsa.com/rsa/S-MIME/) and the MSP (Message Security Protocol) (http://Web.imc.org/workshop/sdn701.ps)
- **Public Key Encryption Standards (PKCS)** from RSA Data Security. These are about using RSA but support other algorithms too. Protocol specifications (final versions) can be found at http://www.rsa.com. Quite a few PKCS specifications have been adopted by IETF as Internet standards, notably the PKCS#7 message format.

2.1 Web components & related security issues

The Web is a complex infrastructure. HARP will capture and classify its major components addressing some of the needs and problems they bring to security and the ways this project will address them.

Standards for document exchange: The typical home page returned to the browser is formatted using *Hypertext Markup Language* (HTML). *Java applets* and *browsers plug-ins* are ways to add new functionality to browsers that support them. The HTML, URL, and HTTP standards are simplifications of the following standards: Standard *Generalised Markup Language* (SGML), *Document Style Semantics* and *Specification Language* (DSSSL), *Hytime* and *Common Command Language* (CCL).

Levels of Security: Web transactions can be secured at three different levels: *above* HTTP, *at the* HTTP level, or *below* HTTP. Securing transactions above HTTP (CCI-PGP) involves the usage of HTTP as a transport mechanism for transferring data that will be decoded by external applications. At the HTTP level, the protocol can be enhanced to deal with encryption and authentication either in an ad-hoc way (SHTTP), or by adding security (SEA) to the protocol using an extension protocol (PEP). Below HTTP a number of protocols (SSL, TLS, PCT, GSS-API, DCE Web, and IPSEC) can be used to establish a secure and authenticated session on top of which the transactions can take place. The protocols and implementations are currently evolving rapidly and a choice among them is usually made depending on the needs of the various applications.

2.2 Integration Technologies

There are various technologies available for integration with the Web, notably CGI-scripts, the Java platform, Active-X, and Javascript. Agents [4] constitute a new technology that also seems promising in a Web-environment. CGI-scripts provide interfaces to other systems on a server, using Web-access, while Java and Active-X additionally supports downloading of code from a server to a client (applets). Agents stretch this even further, by offering code (objects) that can move in a network to accomplish its tasks.
On the other hand the security risks of loading and running unknown code on a client or a server are evident. To protect against these threats, one must either make sure that the code is trustworthy, or make sure that the code runs in an environment where the damage is limited. Code signing, to detect tampering of code and authenticate its source, is the major security measure in Active-X. Code signing is often poorly understood, although in deed it is an important protection measure. Note that only static parts can be signed, so state information cannot be protected this way.
The usual flaw with code signing schemes is that authentication, trust, and authorisation/roles are not separated. Verification of the signature under a trusted certificate is proof of identity. It is not necessarily proof of trustworthiness, nor authorisation for a certain role or certain access rights.
HARP will investigate the threats that these technologies bring, propose countermeasures and develop tools enhancing the TTP solution. In particular HARP will develop TTP functions that can be added to certification service of a TTP so it can digitally sign Java applets and prevent the execution of a malicious external program on a client computer system.

3. Summary and Open Issues

To summarise a security cross-platform based on TTPs will be designed in the HARP project in order to cope with the diversity of the following Web components:

- *"Ordinary" Web-standards/specifications*: HTML, XML, URL, HTTP, plug-ins, Javascipt etc.
- *Protocols/standards/products* which address security (some security built in): SET, C-SET, Java, Active-X, CORBA, some agent platforms, some browsers/servers, etc.
- *Dedicated security standards or well-known specifications relevant to Web*: SSL/TLS, SSH, PCT, GSS-API, DCE Web, IPSEC, PGP, PKCS#7, S/MIME etc.
- *Levels of Security*: e.g., above HTTP, at the HTTP level, or below HTTP).
- *Message security* (build a protected message that can be sent over an insecure network) - supports signature and non-repudiation.
- *Security in major Web applications* (e.g. electronic commerce, telemedical applications).
- *Provision of Secure VPN Trusted Third Parties Services*

4. References

[1] HARP Project, http://telecom.ntua.gr/~HARP/HARP/HARP.htm
[2] Galis A. – "Multi-Domain Communication Management Systems" – CRC Press LLC – ISBN 0-8493-0587-X, July 2000
[3] Murhammer M *et al.*, "A guide to Virtual Private Networks", ITSO series, Prentice-Hall, 1998.
[4] Vigna G (ed.), "Mobile Agents and Security", Lecture Notes in Computer Science, 1419, Springer Verlag, 1998.