Important Security Terminology

By Chris Palmer

Chris is a security engineer on the Chrome Security Team, focusing on secure usability.

By Matt Gaunt

Matt is a contributor to Web**Fundamentals**

TL;DR

- Public/private keys are used to sign and decrypt messages between the browser and the server.
- A certificate authority (CA) is an organization that vouches for the mapping between the public keys and public DNS names (such as "www.foobar.com").
- A certificate signing request (CSR) is a data format that bundles a public key together with some metadata about the entity that owns the key

What are the public and private key pairs?

A **public/private key pair** is a pair of very large numbers that are used as encryption and decryption keys, and that share a special mathematical relationship. A common system for key pairs is the **RSA cryptosystem**. The **public key** is used to encrypt messages, and the messages can only be feasibly decrypted with the corresponding **private key**. Your web server advertises its public key to the world, and clients (such as web browsers) use that to bootstrap a secure channel to your server.

What is a certificate authority?

A **certification authority (CA)** is an organization that vouches for the mapping between public keys and public DNS names (such as "www.foobar.com"). For example, how is a client to know if a particular public key is the *true* public key for www.foobar.com? A priori, there is no way to know. A CA vouches for a particular key as being the true one for a particular site by using its own private key to **cryptographically sign** the website's public

key. This signature is computationally infeasible to forge. Browsers (and other clients) maintain **trust anchor stores** containing the public keys owned by the well-known CAs, and they use those public keys to **cryptographically verify** the CA's signatures.

An **X.509 certificate** is a data format that bundles a public key together with some metadata about the entity that owns the key. In the case of the web, the owner of the key is the site operator, and the important metadata is the DNS name of the web server. When a client connects to an HTTPS web server, the web server presents its certificate for the client to verify. The client verifies that the certificate has not expired, that the DNS name matches the name of the server the client is trying to connect to, and that a known trust anchor CA has signed the certificate. In most cases, CAs do not directly sign web server certificates; usually, there is a **chain of certificates** linking a trust anchor to an intermediate signer or signers, and finally to the web server's own certificate (the **end entity**).

What is a certificate signing request?

A **certificate signing request (CSR)** is a data format which, like a certificate, bundles a public key together with some metadata about the entity that owns the key. However, clients do not interpret CSRs; CAs do. When you seek to have a CA vouch for your web server's public key, you send the CA a CSR. The CA validates the information in the CSR and uses it to generate a certificate. The CA then sends you the final certificate, and you install that certificate (or, more likely, a certificate chain) and your private key on your web server.

Previous
← Why HTTPS

Next
Enabling HTTPS on Your Servers →

Last updated July 2, 2018.