EAP Server Certificate considerations

Almost all EAP types in eduroam (with the exception of EAP-PWD) require an X.509 server certificate with which the RADIUS server identifies itself to the end user before the user sends his credentials to the server.

Consideration 1: Procuring vs. creating your own server certificate

In a generic web server context, server certificates are usually required to be procured by a commercial Certification Authority (CA) operator; self-made certificates trigger an "Untrusted Certificate" warning. It makes sense for browsers to have a pre-configured trust store with many well-known CAs because the user may browse to any website; and the operator of that website may have chosen any of those well-known CAs for his website. In an abstract notion, one can say: it is required to have many CAs in the list because the user device does not have all required information for certificate validation contained in its own setup; it misses the information "which CA did the server I am browsing to use to certify the genuinity of his website?".

These considerations are not at all true in an EAP authentication context, such as an eduroam login. Here, the end user device is pre-provisioned with the entire set of information it needs to verify this specific TLS connection: the IdP has a certificate from exactly one CA, and needs to communicate both that CA and the name of his authentication server to the end user. A trust store list from the web browser is thus insignificant in this context; certificates from a commercial CA are as valid for EAP authentications as are self-made certificates or certificates from a small, special-purpose CA. For a commercial CA, the installation of the actual CA file may be superfluous in some client operating systems (particularly those who make their "web browser" trust store also accessible for EAP purposes), but marking that particular CA as trusted for this specific EAP authentication setup still needs to be done.

Note that also root CA certificates have an expiry date. Both for commercial and private CAs please be aware that an exchange of the root CA certificate will require re-configuration of all your end-users' devices to accept the new CA. As a consequence: for commercial CAs, check their root CA's expiry date so you can make an informed decision whether you want to buy the certificate from them or not. For your own private-use CA: choose a very long expiry date for the CA. Especially for commercial CAs, keep in mind that if you ever want to switch to a *different* CA as a trust anchor, all your end-user devices again need to be re-configured for that new root.

Configuration tools like eduroam CAT enable to provision the chosen CA(s) and the expected server name(s) into client devices without user interaction. In that light, it does not make much difference whether to procure a server certificate from a commercial CA or to make your own; either way, configuration steps are necessary on the end-user device to enable and secure your chosen setup. With the conceptual differences being small, a number of secondary factors come into play when making the decision where to get a server certificate from:

- Do you have the necessary expertise to create a self-signed certificate; or to set up a private Certification Authority and issue a server certificate with it? Consider in particular the next "Consideration 2" which imposes some properties onto the certificates you need.
- Does your eduroam NRO operate a special-purpose CA for eduroam purposes, so that you could get a professionally crafted certificate without much hassle?
- Do your end-user devices all verify the exact server identity (issuing CA certificate AND expected server name)?

The third question is particularly important these days because some popular operating systems, particularly early Android versions up until Android 7, do not allow to configure verification of the expected server name in their UI. For such operating systems, using a commercial CA for the server certificate opens up a loophole for fraud: anyone with a valid certificate from this CA, regardless of the name in the certificate, can pretend to be the eduroam authentication server for your end-user; which ultimately means the end-user device will send the user's login credentials to that unauthorised third-party. If you use a self-signed certificate or private CA however, which issues only one/very few certificates, and over which you have full control, then no unauthorised third party will be able to get a certificate in the first place, and thus can't fraud your users.

Another factor to consider when making the decision private vs. commercial CA is that of size and length of the EAP conversation during every login: with a private CA, you will be able to construct a certificate chain without intermediary CA certificates; requiring less bytes to be transmitted inside the EAP conversation (see Consideration 3, below). This results in fewer EAP round-trips and thus a faster authentication.

So, as a general recommendation: if you have the required expertise, it is suggested to set up a private CA exclusively for your IdP's eduroam service. This CA should have a very long lifetime to prevent certificate rollover problems. The CA should issue only server certificates for your eduroam IdP server (s). If you do not have that expertise, you should make use of your NROs special-purpose CA if it exists. If none of these work for you, a certificate from a commercial CA is the third option.

> ⚠️ *With great power comes great responsibility. (Voltaire)*

If you choose to use a private CA and deploy it to your users' devices, there may be side-effects after the installation of the CA. Some devices do not differentiate between a CA which is used for Wi-Fi server authentication purposes and, say, web browser TLS encryption.

Your CA may incidentally yield the power on such client devices of your own user base to inspect their web or other traffic (if you actively abuse it and modify your IT infrastructure to enable this). We do not endorse or encourage this in any way.

Having your own trusted root CA in client devices also makes the protection of the private key to this CA an objective of paramount importance.

We recommend that you inform your users how best to restrict the power of the CA (e.g. with CA installation instructions which point to a dedicate Wi-Fi store [Android 4.3+]; or with the advice not to use browsers which use the built-in CA store of the device [MS Windows]).

Consideration 2: Recommended certificate properties

Various end-user device operating systems impose different requirements on the contents of the server certificate that is being presented. Luckily, these requirements are not mutually exclusive. When creating or procuring a server certificate, you should check with the CA that its certificates satisfy as many of these requirements as possible to ensure broad compatibility with your users' devices. The list below does not include "standard" sanity checks applied to certificates; e.g. well-formedness of the data, validity timestamps etc. These checks are done "as per usual" in every TLS connection.

The most important property of the server certificate is the name of the server. Since this certificate is not for a webserver, there is no necessity to put an actual hostname into the server name. Also, when an Identity Provider uses multiple servers for resilience reasons, then all these servers can and should have a certificate with the same name; and it may well be the identical certificate. Having different names for different servers means that end-user devices must be configured to trust multiple servers, which is more cumbersome than just having to configure one name string.

Some end-user device operating systems might (incorrectly) require the name to be parseable as a hostname; so it is a good idea to use a server name which parses as a *fully-qualified domain name* - the corresponding record does not have to exist in DNS though. The server name should then be both in certificate's Subject field (*Common Name* component) and be a *subjectAltName:DNS* as well.

The following additional certificate properties are non-standard and are of particular interest in the eduroam context:

Property	Content	Remarks
X.509 version	3	The CA certificate should be an X.509v3 certificate.
server name	parses as fully-qualified domain name	Server certificates with spaces, e.g. "RADIUS Service of Foo University" are known to be problematic with some supplicants, one example being **Apple iOS 6.x**.
server name	Subject /CN == SubjectAltName: DNS	Some supplicants only consult the CN when checking the name of an incoming server certificate (**Windows 8 with PEAP**); some check either of the two; some new EAP types such as **TEAP**, and **Linux clients configured by CAT 1.1.2+** will only check SubjectAltName:DNS. Keeping the desired name in both fields in sync is a safe bet for futureproofness. Only use one CN. If you have multiple RADIUS servers, either use the same certificate for all of them (there is no need for the name to match the DNS name of the machine it is running on), or generate multiple certificates, each with one CN /subjectAltName:DNS pair.
server name	not a wildcard name (e.g "*.someidp.tld")	Some supplicants exhibit undefined/buggy behaviour when attempting to parse incoming certificates with a wildcard. **Windows 8 and 8.1** are known to choke on wildcard certificates.
signature algorithm	Minimum: SHA-256 Recommended: SHA-256 or higher	Server certificates signed with the signature algorithm MD5 are considered invalid by many modern operating systems, e.g. ***Apple iOS 6.x and above***. Also **Windows 8.1** and all previous versions of **Windows (8, 7, Vista)** which are on current patch levels will not validate such certificates. Having a server certificate (or an intermediate CA certificate) with MD5 signature will create problems on these operating systems. Apparently, no operating system as of yet has an issue with the root CA being self-signed with MD5. This may change at any point in the future though, so when creating a new CA infrastructure, be sure not to use MD5 as signature algorithm anywhere. The continued use of SHA-1 as a signature algorithm is not recommended, because several operating systems and browser vendors already have a deprecation policy for SHA-1 support. While the deprecation in browser-based scenarios does not have an immediate impact on EAP server usage, it is possible that system libraries and operating system APIs will over time penalise the use of SHA-1. Therefore, for new certificates, SHA-256 is recommended to avoid problems with the certificate in the future.
length of public key	Minimum: 2048 Bit Recommended: 3072 Bit or higher	Server certificates with a length of the public key below 1024 bit are considered invalid by some recent operating systems, e.g. ***Windows 7 and above***. Having a server certificate (or an intermediate CA certificate) with a too small public key will create problems on these operating systems. The continued use of 1024 bit length keys is not recommended, because several operating systems and browser vendors already have a deprecation policy for this key length. While the deprecation in browser-based scenarios does not have an immediate impact on EAP server usage, it is possible that system libraries and operating system APIs will over time penalise the use of short key lengths. 2048 bit is the most popular and default choice these days. However, some applications already suggest 3072 bit or more, and a longer key length does not have an extra cost. So, it is recommended to create new certificates with 3072 bit keys or higher (4096 has been tested and is also unproblematic) to avoid problems with the certificate in the future.
Extension: Extended Key Usage	TLS Web Server Authentication	Even though the certificate is used for EAP purposes, some popular operating systems (i.e. ***Windows XP and above***) require the certificate extension "TLS Web Server Authentication" (OID: 1.3.6.1.5.5.7.3.1) to be present. Having a server certificate without this extension will create problems on these operating systems.
Extension: CRL Distribution Point	HTTP /HTTPS URI pointing to a valid CRL	Few very recent operating systems require this extension to be present; otherwise, the certificate is considered invalid. Currently, ***Windows Phone 8*** is known to require this extension; ***Windows 8*** can be configured to require it. These operating systems appear to only require the extension to be present; they don't actually seem to download the CRL from that URL and check the certificate against it. One might be tempted to fill the certificate extension with a random garbage (or intranet-only) URL which does not actually yield a CRL; however this would make the certificate invalid for all operating systems which do evaluate the extension if present. So the URL should be a valid one.

Extension: BasicConstraint (critical)	CA: FALSE	Server certificates need to be marked as not being a CA. Omitting the BasicConstraint:CA totally is known to cause certificate validation to fail with **Mac OS X 10.8 (Mountain Lion)**; setting it to TRUE is a security issue in itself. Always set the BasicConstraint "CA" to false, and mark the extension as critical.
Certificate Type	Domain-Validated (DV) or Organisation-Validated (OV)	There have been several reports that **ChromeOS** will refuse to accept Extended Validation (EV) certificates. You should avoid these types of certificates if you care about this operating system.
Validity Time	825 days or fewer	**Apple** products as of **macOS 10.15+ and iOS 13+** enforce this limit and consider certificates with a longer lifetime as untrusted. See also this Apple article.

Consideration 3: Which certificates to send in the EAP exchange

End-user devices need to verify the server certificate. They do this by having a known set of trustworthy anchors, the "Trusted Root Certificates". These root certificates need to be available and activated on the device prior to starting the eduroam login. Therefore, it does not serve any useful purpose to send the root CA certificate itself inside the RADIUS/EAP conversation. It is not harmful to send it anyway though, except that it unnecessarily inflates the data exchange, which means more round-trips during eduroam authentication, and in turn a slower login experience. One possible exception is: there are reports of certain Blackberry devices for which it is advantqageous to send the root CA certificate nontheless; if you expect you need/want to support Blackberry devices, sending the root CA may be of help.

During the EAP conversation, the eduroam IdP RADIUS server always needs to send its server certificate.

One question needs an administrative decision: if there is one or more intermediate CAs between the root CA and the server certificate (such as is the case with, for example, the TERENA Certificate Service (TCS) and many commercial CAs), should the intermediate CA certificates be sent to the end user device during the EAP conversation, or should the devices pre-install the intermediate CAs along with the root certificate?

In any case, for a successful verification of the server certificate, the end-user's device must have the full set of CA certificates available. It does not matter whether the intermediate CAs have been pre-provisioned or are sent during the login phase; but if any one intermediate CA is missing, the verification of the server certificate will fail.

Pre-provisioning the intermediate CAs has the advantage of a relatively small amount of data being sent during the EAP authentication, which means fewer round-trips between the end-user's device and the eduroam IdP RADIUS server. The downsides of this approach are that any changes to intermediate CAs (re-issue, rollover) will also need to be pushed to end-user devices. Also, if end-user devices are not under administrative control of the IdP, there is a danger that some end users do not follow the advice to install all intermediate CAs even though they should, and end up in a situation where the server certificate can not be validated.

Sending the intermediate CAs during the login phase means a longer time to authenticate due to more round-trips, but means that it is sufficient for client devices to install the root CA certificate; if intermediate CAs change, the new ones will always become available to the device during the next authentication data exchange.

For most deployments, it probably makes more sense to include the intermediate CA certificates during the RADIUS/EAP conversation.