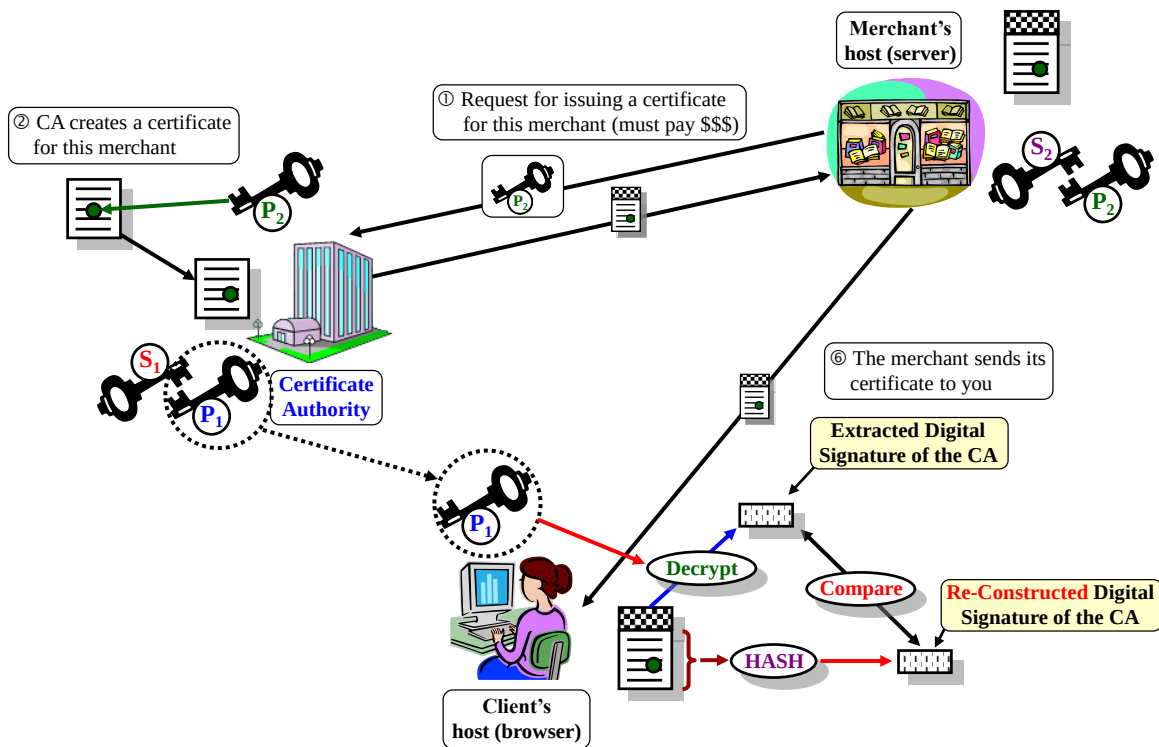


The Root CA's: The anchors for the chain of CAs' trust (for Question #10 in the sample Quiz #6 questions)

1. Your browser needs the public key of the CA

When your browser downloads a digital certificate from a website your browser visited (©), your browser needs the public key of the CA who issued the digital certificate (P_1).



Your browsers should have the public keys of the major existing CAs when you purchase a new PC or when you install a new OS to your PC (the public keys are pre-installed in a new OS).

The question is what will happen if your browser does not have the public key of the CA who created the digital certificate of the website. This can happen if you are using an old PC and the CA started its business after your PC was set up (i.e., when your PC was set up, the CA did not exist).

Since your browser needs the public key of the CA (P_1) for decrypting the digital signature in the digital certificate your browser downloaded from the merchant's website (©), your browser downloads the public key of the CA from the website of the CA. The digital certificate your browser downloaded from the merchant's website includes the contact information of the CA, as well as its URL.

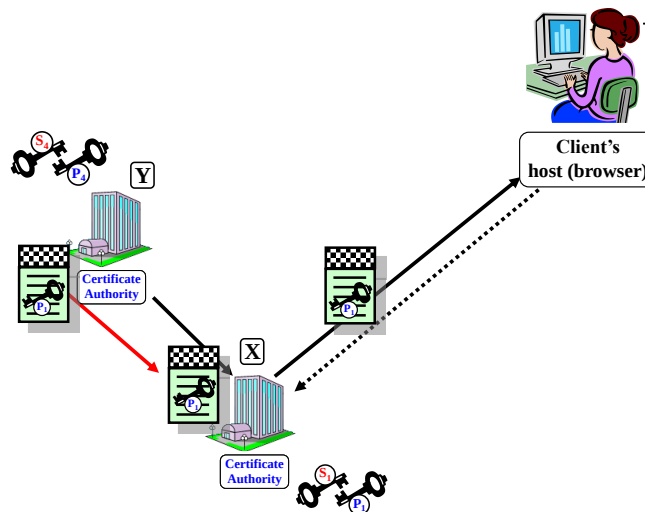
Now, we have a problem. How your browser is sure that the CA is not a bogus CA? Some may say, it should be a legitimate CA because its information is in the digital certificate your browser downloaded from the merchant's website.

Unfortunately, the logic does not prove the legitimacy of the CA because: of the following two reasons:

- ① The certificate posted at the merchant actually came from the same CA (the one in the figure in the previous page of this document).
- ② The CA and the merchant can collaborate (they are both in the same team of this security attack).

2. Our browsers need a mechanism to prove the legitimacy of the CAs (when our browsers are downloading their public keys)

This sounds strange, but our browsers need a help from another CA (let us refer it as “Y”) to confirm the legitimacy of the CA (the CA your browser is downloading its public key (P_1) – let us refer it as “X”). For doing it, another CA (“Y”) issues a digital certificate, which confirms the public key of “X” (P_1).



Here, unfortunately, we face another, but the same problem. Your browser needs the public key of “Y” (P_4) to decrypt the digital signature in the digital certificate Y created for X to prove the legitimacy of the public key of X (P_1). If your browser already has the public key of Y (P_4), this process successfully completes here (your browser successfully confirms the legitimacy of the public key of X through Y).

What will happen if your browser does not have the public key of Y (P_4)? Surprisingly, your browser needs a help from another CA (let us refer it as “Z”). Then, the same process needs to repeat every time your browser bumps up with a CA.

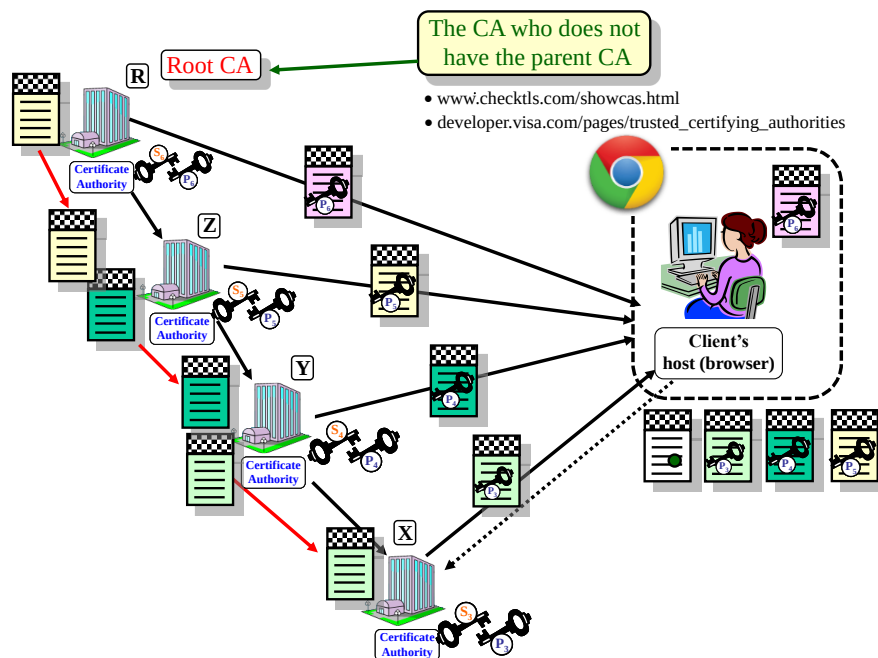
3. We still got a problem ...

After all, it can happen that your browser needs to keep following a chain of the trust among the CAs your browser bumps up with (as shown below).

Finally, we have two questions:

Question 1: Our browsers can never follow an infinitely long chain of CAs. How long our browsers should follow the chain of the CAs' trust?

Answer 1: There are (a small number of) the special CAs, called "root CAs". Those CAs are the ones that do not have any other CAs above them (there is no other CA who proves the legitimacy of the root CAs). Instead, the legitimacy of the root CAs is confirmed by law-enforcement organizations, such as the governments).



Question 2: What will happen if your browser can never reach a root CA after your browser starts following a chain of the CAs' trust (which often happens when your visited a bogus website prepared by an attacker)?

Answer 2: Your browser will give you a warning (please do NOT click "proceed anyway" for your safety):

