Certificate Authorities: Reasoning about cross-linked CA's

Lecturer: Professor Fred B. Schneider
Lecture notes by: Militsa Sotirova, Sam Hinson, Caroline Lui, Lorenzo Scotto di Vettimo, Harry Dang, Rachel Brotherton, Bryan Tantisujjatham, Aliva Das, Kunal Vaishnavi, Dubem Ogwulumba, Grace Jia, Linda Huang

---

A *certificate authority* (CA) stores bindings from principal names to public keys (certificates) in a database that might look like:

$\langle A,\ K_A \rangle k_{CA}$
$\langle G,\ K_G \rangle k_{CA}$
...

Note that each certificate in the database is digitally signed with the certificate authority's private key (as indicated by the $\langle \ldots \rangle k_x$ syntax). This means that anyone with access to CA's public key can check the signature. If we trust signer CA, then we should trust this binding of a principal's name to a public key.

Public key K_{CA} of CA is assumed to be available on every machine, so software running on a machine can verify the signature on any certificate it gets from theCA. Note that no means is being provided for changing K_{CA}, so we are assuming that k_{CA} is never compromised.

We also assume that there is an isolated air-gapped machine (but with access to a power source). This machine stores the sole copy of private key $k_{CA}$and has a mechanism to sign a certificate by using k_{CA}. An authorized operator would access the machine, generate a certificate using the private key stored on the machine, transfer the certificate to some storage device or dongle, and carry that storage device (with the certificate) to the machine storing the CA database. Thus, private key k_{CA}remains isolated on the air-gapped machine.

From a Single CA to Multiple Cross-linked CA's:

It is unrealistic to expect everyone to trust a single CA. But we still must provide means for people in different countries (say, who presumably trust different CA's) to communicate securely.

Suppose there exist multiple CA's, each storing certificates that have been signed with separate private keys. For example, say we have two CA's, one run by CIA and the other by KGB, with the following entries (and CA private keys $k_{CA-CIA}$and k_{CA-KGB} respectively):

CIA:

$\langle A,\ K_A \rangle k_{CA-CIA}$
...

KGB:

$\langle B,\ K_B \rangle k_{CA-KGB}$
...

Suppose A wants to learn B's public key, but B's public key is not in the CIA certificate database (it is only in the KGB database). If A can retrieve $\langle B,\ K_B \rangle k_{CA-KGB}$from the KGBdatabase, how will A know whether to trust that K_B is really B's public key?

Solution: CIA might store a certificate $\langle KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$ that gives a binding for K_{CA-KGB}

CIA:

$\langle A,\ K_A\rangle k_{CA-CIA}$

...

$\langle KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$

KGB:

$\langle B,\ K_B\rangle k_{CA-KGB}$

...

Now, if A wanted to learn B's public key, A would obtain the following *chain of certificates*:

A has K_{CA-CIA}
A retrieves $\langle KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$ from the CIA database (because A has K_{CA-CIA}, she knows she can trust this certificate)
A retrieves $\langle B,\ K_B\rangle k_{CA-KGB}$ from the KGB database
A uses K_{CA-KGB} to check that she can trust $\langle B,\ K_B\rangle k_{CA-KGB}$
Now A can use K_B

(Note that A still must trust KGB to be correctly reporting B's public key.)

Such a chain of certificates can seem complicated. Therefore, we might seek some formal way of justifying that, given some collection of known information, we should trust the conclusions we're being asked to draw from the certificate chain.

How do we know we can trust the result of this chain of certificates?

The important insight is that we will associate a principal with each public key. Some definitions:

- Principals can **say** things, e.g. P**says** m. Principals only say things that they believe, so we posit a set $\omega(P)$ defined to be the set of statements (propositions) that P believes. $\omega(P)$ is also known as the *worldview* of P. So P**says** m holds if and only if $m \in \omega(P)$ holds.

- Principals may "speak for" (**sfor**) other principals (e.g. a keyboard on which A is typing is a principal and that principal speaks for A). Henceforth let A **sfor** B mean that A "speaks for" B (although note that in some literature, the notation $A \Rightarrow B$ is used instead) Formally, we will say that A **sfor** B holds if and only if $w(A) \subseteq w(B)$ holds. Intuitively, this means that A cannot say things that B cannot say (but B may say things that A cannot). So, anything that A says, B says, too.
  - Slightly modifying our keyboard example, a keyboard in practice will say things that are completely independent of A, and thus, not a part of A's worldview. In this more realistic setting, the set of beliefs within a keyboard's worldview would *not* be a subset of those of A, and thus cannot speak for A.

- Finally, denote the following notation will be used to give the inference rules that allow us to reason about formulas involving **says** and **sfor**:

If we have proved or can assume hypotheses $H_1,\ ...,\ H_n$, then we can conclude : $\dfrac{H_1, ..., H_n}{C}$

Uses of inference rules formalize logical reasoning, in a way that enables assurance about a conclusion. Such rules are not only used in formal logics (e.g., propositional logic and predicate logic) but they are used as typing rules for compilers, statistical learning, etc.

Formal Inference Rules for **says** and **sfor**:

R1. $\dfrac{\langle m \rangle_k}{K \textbf{ says } m}$

If a message m is signed by k, then (by convention) K **says** m holds. So the crypto key-pair (K, k) is being viewed as defining a principal, and this principal is identified by its public key K.

R2. $\dfrac{A \textbf{ sfor } B,\ B \textbf{ sfor } C}{A \textbf{ sfor } C}$

This follows from the transitivity of the $\subseteq$ operator. If $\omega(A) \subseteq \omega(B)$ (A speaks for B) and $\omega(B) \subseteq \omega(C)$ (B speaks for C), then transitivity of the $\subseteq$impliest be that $\omega(A) \subseteq \omega(C)$ (A speaks for C).

R3. $\dfrac{A \textbf{ sfor } B,\ A \textbf{ says } X}{B \textbf{ says } X}$

This rule can be justified as follows. From hypothesis A **sfor** B we have that $\omega(A) \subseteq \omega(B)$ holds. From hypothesis A **says** X, we have that $X \in \omega(A)$. So it follows that $X \in \omega(B)$, which is the meaning (see above) of B**says** X.

R4. $\dfrac{A \textbf{ says } B \textbf{ sfor } A}{B \textbf{ sfor } A}$

Intuitively, A is being considered the authority on the contents of $\omega(A)$. If A says that someone (B) speaks for her, then we believe A, and we conclude that B speaks for A. Concrete example: let $A = FBS$ and $B = TA$. If FBS says that TA speaks for FBS, then because FBS believes that $\omega(TA) \subseteq \omega(FBS)$, we assume that FBS should know whether this is really true or not, and trust that the TA does in fact speak for FBS.

R5. $\dfrac{A.n \text{ is a subprincipal of } A}{A \textbf{ sfor } A.n}$

A subprincipal inherits the views of its uber-principal, and may have more views (formally, $\omega(A) \subseteq \omega(A.n)$).

Example Proof

Let's return to the example from before, where A is trying to learn the public key of B.

To attach a story to this example: Alice is a CIA agent who has fallen in love with Boris, a KGB agent. Alice and Boris would like to communicate with one another in an encrypted manner. Suppose that Alice wishes to send a message to Boris. She must learn Boris's public key K_B. This key is not maintained by the CIA's Certificate Authority but rather is managed by the KGB's. Thus, Alice must communicate with the KGB's Certificate Authority. As we have seen above, this requires learning K_{CA-KGB}. Since Alice works for the CIA, she only knows the public key for the CIA's Certificate Authority, K_{CA-CIA}. We assume, however, that the CIA maintains a record for K_{CA-KGB}.

Alice knows:

- K_{CA-CIA} **sfor** $CA-CIA$ (aka, K_{CA-CIA} is the public key for $CA-CIA$)

Alice receives:

- From the CIA's Certificate Authority: $\langle CA-KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$
- From the KGB's Certificate Authority: $\langle B,\ K_B\rangle k_{CA-KGB}$

First, assume that we can treat B as a subprincipal of the KGB with the notation: $KGB.B$. Then, formally, we can create a proof tree using the above inference rules as follows:

$$\dfrac{\dfrac{\dfrac{\langle KGB.B, K_B\rangle k_{CA\text{-}KGB}\ (R1)}{K_{CA\text{-}KGB}\ \textbf{says}\ K_B\ \textbf{sfor}\ KGB.B}\quad\dfrac{\langle KGB, K_{CA\text{-}KGB}\rangle k_{CA\text{-}CIA}\ (\langle \text{Assumption}\rangle)}{K_{CA\text{-}KGB}\ \textbf{sfor}\ KGB}\ (R3)}{KGB\ \textbf{says}\ K_B\ \textbf{sfor}\ KGB.B}\quad\dfrac{(R5)}{KGB\ \textbf{sfor}\ KGB.B}\ (R3)}{\dfrac{KGB.B\ \textbf{says}\ K_B\ \textbf{sfor}\ KGB.B\ (R4)}{K_B\ \textbf{sfor}\ KGB.B}}$$

Note the $\langle$Assumption$\rangle$ in the proof tree, in order to conclude that $\langle KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$ means that K_{CA-KGB} must speak for KGB. Essentially, we decide that we trust the CIA to be an authority on the public keys of entities. We will revisit this assumption later.

To explain the above proof tree:

1. Alice receives: $\langle KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$
    a. This is equivalent to: K_{CA-CIA} **says** "K_{CA-KGB} **sfor** KGB"
2. Next, Alice goes to the KGB's CA and asks for Boris's public key. She receives: $\langle KGB.B,\ K_B\rangle k_{CA-KGB}$ **(R1)**
    a. Note in this notation that we are treating Boris as a subprincipal of KGB. This will allow us to apply R5 later.
    b. Similar to before, this is equivalent to: K_{CA-KGB} **says** "K_B **sfor** $KGB.B$"
3. We would like to show that K_B **sfor** $KGB.B$, as that implies that what K_B says/believes is what $KGB.B$ also believes (aka, K_B is Boris's public key). Unfortunately it is not obvious how to proceed. However, we will make an assumption and that will allow us to make progress. We will later revisit the implication(s) of this assumption.
4. For now, let us assume:
    a. K_{CA-KGB} **sfor** KGB **(assumption)**
5. If we have this we can use R3 to substitute KGB for K_{CA-KGB} in any statements K_{CA-KGB} makes, specifically statement 2b. This gives:
    a. KGB **says** "K_B **sfor** $KGB.B$" **(R3)** (Comes from K_{CA-KGB} says "K_B **sfor** $KGB.B$")
6. Since we are treating Boris as a subprincipal of KGB we know via R5 that:
    a. KGB **sfor** $KGB.B$ **(R5)**
7. Again, we use R3 to substitute $KGB.B$ for KGB in the result from step 5 (KGB **says** "K_B **sfor** $KGB.B$").
    a. $KGB.B$ **says** "K_B **sfor** KGB.B" **(R3)**
8. We currently have: "Boris says that K_B speaks for Boris". We now have a simple application of R4:
    a. K_B **sfor** $KGB.B$ **(R4)**

So our assumption paid off! Why is this conclusion useful? Because when Alice receives some message signed by K_B she can use the following reasoning:

1. Alice received $\langle m\rangle k_B$. This is equivalent to K_B **says** m **(R1)**
2. If K_B **sfor** $KGB.B$ we have: $KGB.B$ **says** m **(R3)**, as desired

Finally, let us revisit the assumption. We assumed K_{CA-KGB}**sfor** KGB, aka that the key pair Alice received from the CIA's CA is accurate. From the first thing that Alice knows, we have: K_{CA-CIA} **says** "K_{CA-KGB}**sfor** KGB".

Let us assume that KGB is a subprincipal of CIA. Then, by **(R5)**, CIA **sfor** KGB. So we have "CIA **says** K_{CA-KGB}**sfor** KGB", we can conclude that this means "KGB **says** K_{CA-KGB} **sfor** KGB".

To elaborate:

1. Alice Receives:$\langle KGB,\ K_{CA-KGB}\rangle k_{CA-CIA}$
    a. Which means K_{CA-CIA}**says** "K_{CA-KGB} **sfor** KGB"
2. And since Alice knows that K_{CA-CIA} is the public key for the CIA (aka that K_{CA-CIA} **sfor** $CA-CIA$), we can use **(R3)** to get:
    a. CIA **says** "K_{CA-KGB} **sfor** KGB"
3. If we can assume that KGB is a subprincipal of CIA, then we can use **(R5)** to transform this statement into:
    a. KGB **says** "K_{CA-KGB} **sfor** KGB"
4. Then, by **(R4)**, we have:
    a. K_{CA-KGB} **sfor** KGB
5. So we have succeeded in showing that K_{CA-KGB} **sfor** KGB if KGB is a subprincipal of CIA. However, if KGB wasn't a subprincipal of CIA and CIA couldn't speak for KGB, then this proof wouldn't work.

Thus, this analysis has revealed a crucial assumption that might otherwise have gone unnoticed if A's analysis of the keys and certificates she received, in order to decide whether she trusts that K_B speaks for B, was more informal.