Team Rap: Rasmus Jørgensen (201909451), Mathias Weller (201907948) & Anna Blume (201907691)

Hand-in 4 (DISSY)

6.1

You are consultant to a company that offers access to a database D. Different users have access to different parts of the database and moreover, the database contains sensitive information. Therefore, the security policy says:

- When a request for information arrives, D should be able to determine which user sent the request.
- A user should not be able to get information about which data other users asked for.

We can assume that every user A has a private RSA key sk_A of his own, and the database stores a list of public keys of all users. Also, all users know the public key pk_D of D.
The threat model assumes that the database is sufficiently protected that it will not be hacked, but network traffic can be attacked and manipulated.
One suggestion to ensure that the policy is followed is: when user A wants to send a request R to D, he will send $E_{pk_D}(R), S_{sk_A}\left(E_{pk_D}(R)\right), A$ to D, that is, encrypt the request under D's public key, append his signature on the encrypted request, and then append his username. D will look up pk_A, verify the signature, and if it is correct, will decrypt the request, and return an answer of form $E_{pk_A}(data)$, that is, the answer to the request, encrypted under A's public key.
Another suggestion goes as follows: when user A wants to send a request R to D, he will send $E_{pk_D}\left(R, S_{sk_A}(R)\right), A$ to D, that is, append his signature on the request, encrypt this under D's public key, and then append his username. D will decrypt the request and signature, look up pk_A, verify the signature, and if it is correct, will return an answer of form $E_{pk_A}(data)$, that is, the answer to the request, encrypted under A's public key.
One of these solutions fails to satisfy the security policy. Which one, and why? Are there any general conclusions one could draw from this example?

Den første løsning, altså $E_{pk_D}(R), S_{sk_A}\left(E_{pk_D}(R)\right), A$, fejler. Problemet er at en adversary kan opsnappe beskeden, smide signeringen væk og tilføje sin egen signering af $E_{pk_D}(R)$ på beskeden, samt sit eget navn A' på, den nye besked: $E_{pk_D}(R), S_{sk_{A'}}(R), A'$
Når adversary opsnapper beskeden, ved han selvfølgelig ikke hvad den handler om, men når databasen modtager den modificerede besked med A' som afsender, kan databasen ikke se at A' har ændret afsender og signering, og vil derfor sende svaret på forespørgslen tilbage til A', hvorved A' får svaret på A's spørgsmål. Hvis vi omvendt først signerer og derefter enkrypterer, kan en adversary intet gøre for at ændre beskeden, da både besked og signatur er "beskyttet" af krypteringen.
Hvad vi generelt kan se, er at en signatur kun er sikker hvis det ikke er muligt for en adversary at lave sin egen signatur på det sendte og sende den med i stedet for den oprindelige signatur. Dette kan for eksempel ske ved (som i forslag 2) at kryptere signaturen sammen med beskeden, så de ikke kan skilles ad af en adversary.

Team Rap: Rasmus Jørgensen (201909451), Mathias Weller (201907948) & Anna Blume (201907691)

6.10

For at køre koden:
”go run .” i en terminal.

Assume you had to process the entire message using RSA. Use the result from question 3 to compute the speed at which you could do this (in bits per second). Hint: one of the RSA operations you timed in question 3 would allow you to process about 2000 bits. Compare your result to the speed you measured in question 2. Does it look like hashing makes signing more efficient?

Vi prøver at hashe 80000 bits (10kB). I 3) kan vi processere 2000 bits på ca. 10 millisekunder (ud fra at køre den mange gange). Hele beskeden er $\frac{80000}{2000} = 40$ dele. Så vi kan processere hele beskeden på $40 \cdot 0{,}01$ sekunder. Det giver $\frac{80000}{40 \cdot 0{,}01} = 2 \cdot 10^5$ bits/sekund. I 2) kunne vi processere ca. $1{,}5 \cdot 10^9$ bits/sekund når vi hasher. Så det er hurtigst at hashe først, og derefter signere de 256 bits som bliver hashet, i stedet for at signere hele beskeden.