Jessica Noel 11/23/2020

CS306: Introduction to IT Security (Fall 2020) Homework #2

# Problem 1: Domain-extension MAC implementation (30%)

Given the provided support code in Java, implement a secret-key message authentication code that employs only a block cipher (and no other cryptographic primitive) to authenticate messages of any size in a *bandwidth-efficient manner*. In particular and as specified in the provided instructions:

1. Implement the mac() and verify() methods.
2. Demonstrate that they are correct by providing the MAC tag (in hexidecimal) of the specified default message using the specified default key.
3. Explain which algorithm you implemented and why.
4. Explain what are the domain-extension features of your algorithm in relation to its security.

Hint: Does your implementation securely handle messages of fixed size, messages of any size or messages of any fixed size?

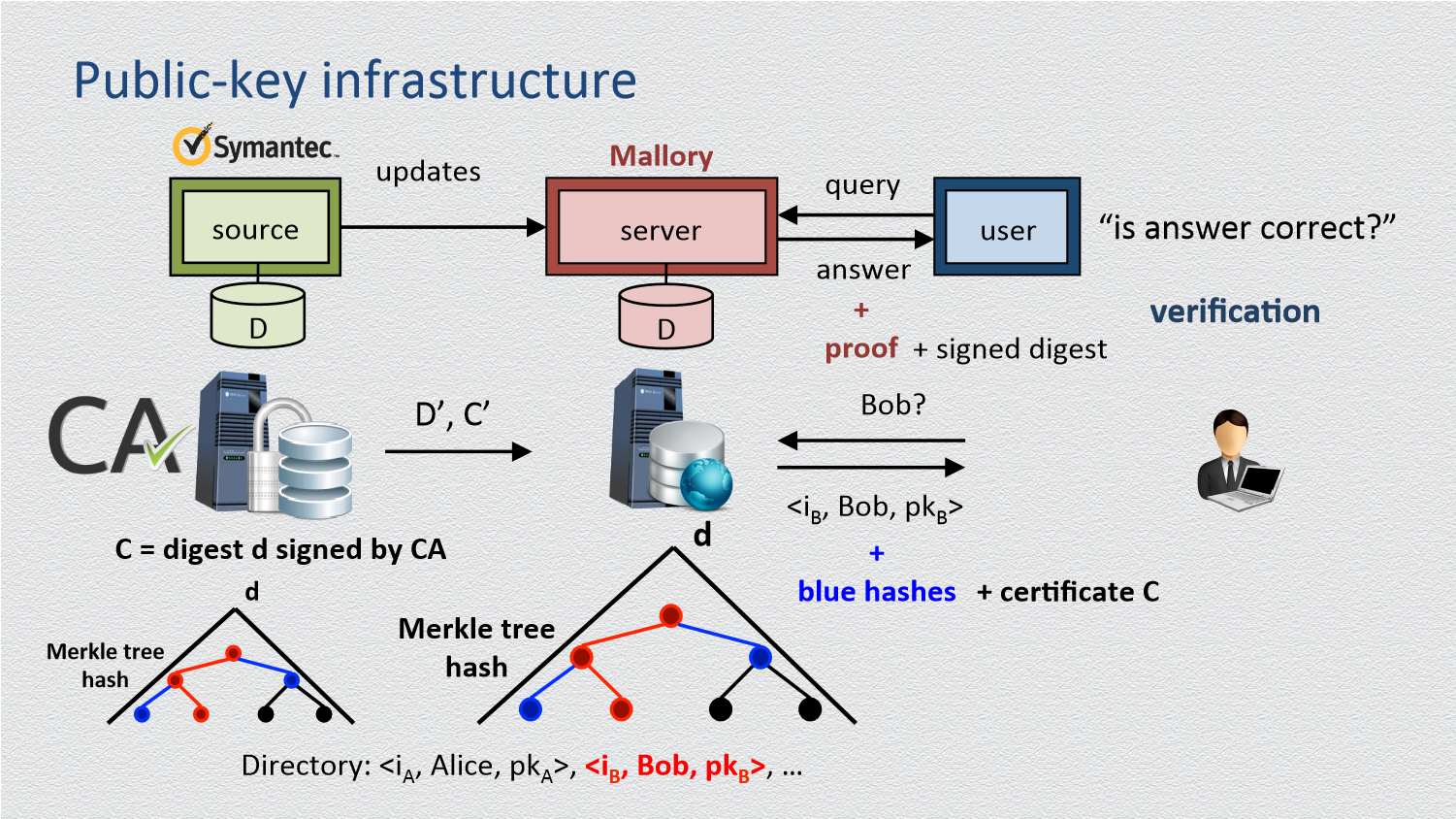


Figure 1: The public-key dictionary-as-a-service model for verifying public keys.

# Problem 2: Data outsourcing & public-key infrastructure (30%)

**(1)** To protect the secrecy of course-related communications, CS306 makes use of public-key encryption: Enrolled students and staff members have their public keys registered with a trusted certification authority (CA), e.g., Symantec; that is, each CS306 person with Stevens UID *i* and name *ni* has a public key pair (*ski,pki*). For efficiency reasons, the CA makes the directory *D* = {(*i,ni,pki*)|*i* ∈ CS306} of all such public keys available (for people to query) through a Stevens online service that is administered by Mallory. Specifically (see also Figure 1):

* The CA provides Mallory with the public-key directory *D* along with a special certificate *C* that is the Merkle-tree digest of the directory signed by the CA.
* To send a confidential message to Bob, Alice asks Mallory for his public key—even if Alice had recently learned his public key via a previous query to Mallory, since public-key pairs can be occasionally refreshed or revoked.
* Along with Bob’s public-key record (*iB,*Bob*,pkB*) in *D*, Mallory also returns to Alice the certificate *C* and a Merkle-tree proof corresponding to Bob’s record.
* After any change in the class enrollment (e.g., a student drops it or enrolls in it with delay) or any key pair is refreshed, the CA provides Mallory with the new (that is, updated) directory *D*0 and the new (that is, corresponding to *D*0) certificate *C*0.

Suppose that Eve manages to secretly get access to Bob’s laptop and successfully steal its secret key *skB*. When Bob becomes suspicious of this, he registers a new public-key pair with the CA.

How can Eve collaborate with Mallory in order to decrypt all subsequent messages sent to Bob? What is the name of this attack type?

**The name of this type of attack is a man-in-the-middle attack. Eve can request Bob’s public key through Mallory, by doing this Eve can see messages being sent to Bob. Eve can now decrypt all sent and received messages having Bob’s private key.**

**(2)** Describe how the use of periodically timestamped signatures (i.e., signatures on a timestamped message) can be employed by the CA to provide a solution to the above attack. You can assume that no public key will be updated twice within the same day, and thus consider a 1-day period.

**By periodically time stamping signatures the CA can provide a solution to the above attack. This is because the timestamping mechanism allows to prove the integrity of a series of data. This meaning that data that was present in a specific moment and has not been altered since then. Timestamping the messaged ensures Eve can not resend any messages from a prior state. The CA will be able to see if there are changes to the private key and notify Bob of them to prevent the attack.**

# Problem 3: Intrusion resilience (30%)

Honeywords comprise a recently proposed method for hardening the password security against stolen password files (after an attacker compromises an authentication server). The idea is to distribute password verification across two servers, say one red and one blue, each storing and verifying “half” of the credentials needed to verify in order to successfully authenticate a user (see also Figure 2). Whereas the final accept/reject decision depends on both the red and the blue partial decision, compromising any one server alone provides no (or little) advantage to an attacker.

1. How does this split architecture improve security? Consider the two cases where an attacker compromises only one of the servers.

**This split architecture improves security because if an attacker were to successfully get into one of the servers this would mean he only has half of the credentials and he needs both in order to have succeeded in a full attack. This system allows not as much information to be leaked in one breach.**

1. Explain how honeywords make password cracking detectable.

Hint: Users are not aware of the existence of honeywords.

**Honeywords are false passwords which are put into a web-site’s database. Most passwords in these databases are hashed for the users protection. Honeywords make password cracking detectable because when a hacker takes the hashes passwords and inverts the hash they are unable to tell if what they found is a legitimate user password or a honeyword. If the hacker is the use the honeyword to log into a user’s account an alarm is set off. This alarm/alert will be seen by an administration who will lock the account to protect.**

1. What constitutes a good honeyword for a user whose real password is pa$$word5, if honeywords are generated by tweaking real passwords, i.e., by keeping the main password structure but changing special symbols and numbers?

**A good honeyword password will be similar to a user’s real password/typical password habits as stated above. Since there are infinitely many ways to tweak a password (as simple as changing one letter or number) and good alternate to use for the password pa$$word5, could be pa$$word6 or pas$word5 (any small tweak to the original password would suffice).**

1. You just stole the honeywords list of one of the employees in the Office of the Registrar, which consists of passwords: Blink-123, Blink-182, itWb!%s45 3gMoI00286!\*mooewTi409##21jUi, and you have only one chance to impersonate him/her (and try to increase your GPA). Which password will you choose and why?

**If I were to choose a password out of the three, I would most likely choose “Blink-182”. This is because it is more likely that Blink-123 is used as a fake password variation than Blink-182. The other password is too lengthy and has a variety of symbols, letters and number making it difficult to see what to change within the honeyword to crack the actual password. It is also a common band, and maybe someone just really likes it, but was silly enough to make such a simple password.**

# Problem 4: On the RSA cryptosystem (30%)

1. The RSA cryptosystem relies on modular exponentiations, as its core operations. How are such operations realized more efficiently in practice? How is RSA decryption/signing further accelerated?

**Modular exponentiations are realized more efficiently in practice when the person implementing the cryptosystem making the cipher text better and well as brute force key. We can also implement a hybrid RSA system which uses two different keys (a private and a public) for decryption and encryption processes. The key values are determined by the value M holds which is the product of four prime numbers. By using this it increases the factorizing of M. This computation of keys allows a more secure encryption and decryption.**

**RSA decryption/signing is further accelerated because it uses modular exponentiation, which is function that is difficult to invert as it has one-way function behavior.**

1. Given the support code in Python that was provided in Lab#8, implement the RSA keygeneration algorithm. Namely, submit the completed skeleton code for Lab#8, including the methods you finished during the lab and the RSA key-generation algorithm, i.e., method keygen(size).

