Cryptography with OpenSSL – Applications

Lab Report 3

Information Systems Security course (01TYM, 02KRQ)
Politecnico di Torino – AA 2023/24

prepared by:

Anuar Elio Magliari (s317033@studenti.polito.it)

George Florin Eftime (s303483@studenti.polito.it)

Alekos Interrante Bonadia (s319849@studenti.polito.it)

1 Cryptography applications: integrity

1. Message Authentication Codes

Manual generation of a keyed-digest

- Write down the command dgst of OpenSSL that Alice could to calculate a keyed-digest on the file msg, by using sha256 as base function?
  - openssl dgst -sha256 -hmac key -out msg_digest -hex msg
- What information needs to know Bob to verify this manually constructed keyed-digest?
  - The original plaintext
  - The base function used for the hashing
  - The symmetric key
- Which steps must be performed by Bob to verify the keyed-digest?
  - Once Bob receives the message from Alice, he should calculate the keyed digest by providing the hash function with the symmetric key and the received message. Afterward, using the resulting hash value, he can compare it with the digest provided by Alice and attached to the received message.
- Why this way of constructing a keyed-digest is not secure?
  - An attacker, even if he doesn't know the key, could append message blocks to the end of the message (or at the beginning, depending on where the key is located) and recalculate the keyed digest.

Computing an HMAC with OpenSSL

- Write the OpenSSL command used to verify the keyed-digest generated above (hint: to compare two files you can use the command cmp)
  - openssl dgst -sha256 -binary -hmac key -out msg.hmac msg
  - openssl dgst -sha256 -binary -hmac key -out msg.hmac_bob msg

- cmp msg.hmac msg.hmac_bob (any output if they are equal)

- Is the HMAC that you have just recalculated different from the one you have calculated previously on the original message?
  - openssl dgst -sha256 -binary -hmac key -out msg.hmac_recalculated msg
  - cmp msg.hmac msg.hmac_recalculated
    - Output: msg.hmac msg.hmac_recalculated difference: byte 1, row 1
    - It's different because the message is different, and consequently, the keyed digest value is also different (excluding possible collisions).
- Assuming you use a secure hash function resistant to collisions, could an attacker modify the message msg and then recalculate correctly an HMAC so that its modification cannot be detected by Bob?
  - If we assume that the hash function is collision-resistant and the attacker does not know the key, it is highly improbable for him to modify only the message while expecting that Bob cannot detect a difference between the calculated HMAC and the HMAC received from Alice.

2. Digital signature

- Write down the dgst OpenSSL command to verify the signature:
  - openssl dgst -sha256 -verify ./alice/rsa.pubkey.alice -signature chap11.sig chap11.pdf
- Do you note any difference when you use the pkeyutl command instead of the dgst command?
  - For the calculation: with the pkeyutl command, we specify the key (Alice's public key) to sign the hash, whereas in this case, with the dgst command, we do not specify any key (protection of keyed digest with an asymmetric key).
  - For the verification: with pkeyutl command, it’s necessary to specify also the hash file such that to compare the file decrypted “chap11.sig” and compare it with the hash “chap11.hashbob” while with the dgst command is enough to specify the algorithm to calculate the digest from the original file “chap11.pdf” and once decrypted the file “chap11.sig”, they can be compared each other.
    - openssl pkeyutl -verify -in chap11.hashbob -sigfile chap11.sig -inkey rsa.pubkey.alice -pubin
    - openssl dgst -sha256 -verify ./alice/rsa.pubkey.alice -signature chap11.sig chap11.pdf
- What happens if the attacker Chuck modifies one of the data blocks Alice sent to Bob? In practice, if Chuck modifies the original message (but not the signature), which will be the result of signature verification?
  - If the attacker only modifies the message, Bob will calculate the hash (as specified in the procedure). When he verifies the signature, he will detect that the original message has been altered during transmission because the calculated hash and the hash decrypted from the signature will not match.

- If Chuck modifies the signature (but not the data), which will be the result of signature verification?
    - If only the signature is modified, Bob will decrypt the signature to obtain the hashed message (which he assumes was signed by Alice). However, when he compares the locally calculated hash message with the hash decrypted from the signature, the comparison will fail.
- What happens if Chuck replaces Alice’s public key with his own public key and sends it to Bob claiming that it’s Alice’s public key?
    - Bob cannot decrypt the signature successfully, because the message was signed with secret key of Alice and only with the corresponding public key Bob can decrypt the message.
- Can he replace the original data Alice created and the signature (created by Alice)? What will be the result of the signature verification in this case?
    - If the attacker replaces Alice's message and signature with their own message and signature, Bob can detect that the message was altered during transmission if he uses Alice's original public key. Otherwise, he won't have the knowledge to detect the difference.

3. Applications of asymmetric cryptography: key exchange/agreement

- Suppose Alice and Bob want to exchange data encrypted with AES, but they do not share a secret key. Can you design a solution based on the OpenSSL commands that you have used so far (for symmetric and asymmetric operations) so that Alice can send an AES-encrypted file to Bob even if Alice and Bob do not share a secret key? (pay attention: Bob must be able to correctly decrypt the ciphertext received from Alice).
    - Alice can exploit asymmetric operations to send the key along with the message. In this case, she can use Bob’s public key to encrypt the key and attach the result to the message. When Bob receives this tuple, he can decrypt the result of the asymmetric operation with his own private key, obtaining the symmetric key, and then successfully decrypt the message.
- Which serious errors did (inexperienced) Alice? (There are at least two!)
    1. She sent the own private key to Bob.
    2. She had signed the symmetric key with her own private key, instead of using the Bob’s public key, so everyone can read the content by using her public key and obtain the symmetric key.
    3. She didn’t send a keyed digest in order to check the integrity of the file.
    4. She encrypted the file with an IV equal to a block of 0s and it’s not sent to Bob, so he must try all possible IVs in order to discover the original IV applied to the encryption (if it’s not inserted in the encryption in same way).
    5. She should use a higher key length (256 bits are safer than 128 bits, it depends for how long it needs to remain secure).
- Have Bob managed to successfully recover the original plaintext (chap12.pdf)?
    - Yes, he recovered the original plaintext successfully

- What kind of security attack could do Chuck to intercept the data messages exchanged between Alice and Bob?
  - The attacker can pretend to be Bob with Alice (and to be Alice with Bob) applying a man-in-the-middle attack, he can intercept the public key sent from Bob to Alice and send his public key (the attacker) to Alice (and vice versa with Bob) so that Alice will encrypt all data with the Chuck's public key (and same for Bob).
- How can you avoid this attack? (Hint: think about the security of the first step in the workflow above.)
  - This attack can be avoided by using the Bob's digital certificate which specifies (and certifies) the correspondence between Bob's identity and his public key (and same for Alice to Bob).
- Which other asymmetric algorithm could Alice and Bob use instead of RSA to perform the key exchange/agreement?
  - In order to agree to symmetric key, Alice and Bob can use the Diffie-Hellman algorithm, which once chosed two numbers p > g > 1, they can exchange the computed key each other.
- Considering that you identified the alternative asymmetric algorithm (instead of RSA) for key exchange/agreement, what kind of security attack could (still) do Chuck to intercept the data messages exchanged between Alice and Bob?
  - Chuck, in this case, can perform a man-in-the-middle attack where he pretends to be Bob with Alice and vice-versa. Chuck will send his key (g^z mod p) to Alice, after intercepting the Bob's key (and vice-versa with Bob).
- How can you avoid this attack?
  - With the use of a digital certificate, this can be avoided. When Alice calculates her key, she can sign the data with her private key as specified in her digital certificate and send the signed key along with her digital certificate..

4. Digital certificates and public key infrastructures

- Is this certificate a valid one ?
  - Validity Not Before: Oct 7 10:09:07 2020 GMT Not After : Oct 7 10:09:07 2021 GMT
    - So, it is not valid, because it's expired.
- How many certificates are in the certificate chain (except the certificate for www.polito.it)?
    - There are 2 other certificates.
  - openssl s_client -showcerts -connect www.polito.it:443 (cmd that show the certificates chain):
  - 2. Intermediate Certificate:Subject:

    Subject:C = NL, O = GEANT Vereniging, CN = GEANT OV RSA CA 4

    Issuer: C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority

Public Key: RSA, 4096 bits

Signature Algorithm: RSA-SHA384

Validity: From Feb 18 00:00:00 2020 GMT to May 1 23:59:59 2033 GMT

- 3. Root Certificate:

  Subject: C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority

  Issuer: C = GB, ST = Greater Manchester, L = Salford, O = Comodo CA Limited, CN = AAA Certificate Services

  Public Key: RSA, 4096 bits

  Signature Algorithm: RSA-SHA384

  Validity: From Mar 12 00:00:00 2019 GMT to Dec 31 23:59:59 2028 GMT

- Who is the issuer of the certificate for the site www.polito.it (write down the entire Distinguished Name, is in the form CN= ..., O = ..., L= ..., ST = ..., C = ...)?
  - Issuer: C = NL, ST = not specified, L = not specified, O = GEANT Vereniging, CN = GEANT OV RSA CA 4
- Which is the subject of the certificate of the site www.polito.it (write down the entire Distinguished Name (C=...,)?
  - Subject: C = IT, ST = Torino, L = not specified, O = Politecnico di Torino, CN = www.polito.it
- Which is the time validity for the certificate of www.polito.it?
  - Not Before Tue, 07 Feb 2023 00:00:00 GMT and Not After Wed, 07 Feb 2024 23:59:59 GMT
- Indicate the issuer, the subject, and time validity of the Root CA certificate in the certificate chain of www.polito.it (that is the last certificate in the chain starting from the certificate for www.polito.it)
  - Issuer: C = GB, ST = Greater Manchester, L = Salford, O = Comodo CA Limited, CN = AAA Certificate Services
  - Subject: C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authority
  - Validity: From Mar 12 00:00:00 2019 GMT to Dec 31 23:59:59 2028 GMT
- Where can be found the Certificate Revocation List (CRL) for the certificate www.polito.it?
  - In CRL endpoint we can download “GEANTOVRSACA4.crl” from this link:
    - http://geant.crl.sectigo.com/GEANTOVRSACA4.crl
  - View the contents with this command:
    - openssl crl -inform DER -in GEANTOVRSACA4.crl -noout -text
- Where it can be found the OCSP responder that may provide the status of the certificate for www.polito.it at current time?

- http://geant.ocsp.sectigo.com/

5. Key exchange with DH

- No questions found.

6. Asymmetric challenge response authentication

- Is the response to the challenge correct?
  - Yes
- Is this method subject to replay attack (explain why)?
  - No, because the challenge is a nonce and a random number so it will be different during every authentication. For each challenge it is accept only one response.
- Is this method subject to a sniffing attack (explain why)?
  - No, because even if the challenge will be sent in clear, it’s a nonce so the next time it will be another different value.
- What if Bob is an attacker? What kind of attack could he do with the challenge and the response received from Alice (hint: we assume that Alice has not authenticated Bob)?
  - Bob, acting as an attacker, might position himself as a man-in-the-middle between Alice and the legitimate verifier. In this scenario, Bob could intercept the challenge resolved by Alice, forward it to the legitimate verifier, and potentially impersonate Alice in communication with the verifier. This could enable Bob to manipulate the data or even impersonate the legitimate verifier when communicating with Alice.