

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology



CRYPTOGRAPHY AND NETWORK SECURITY

Lab 5

Nguyen Phuong Vuong - 1614186

Questions.

1. What types of attacks are addressed by message authentication?

- Brute force attack.
- Disclosure.
- Traffic analysis.
- Masquerade.
- Content modification.
- Sequence modification.
- Timing modification.

2. What are some approaches to producing message authentication?

- Ta dùng cipher-text. 1 giải thuật mã hoá E thực hiện mã hoá message M bằng key K tạo ra 1 cipher-text hợp lệ.
- Sử dụng Tags. 1 tag được đính kèm vào message sử dụng giải thuật tag-generation. Khi nhận được message, có 1 giải thuật tag-verification được áp dụng vào message để xác nhận nó hợp lệ.
- Sử dụng MAC (Message Authentication Code). 1 giải thuật tag-verification không cần thiết ở đây. Người nhận chỉ cần thực hiện công thức $T = \text{MAC } k(M)$ để xác nhận nó là hợp lệ.

3. What is a message authentication code?

- Nó còn được gọi là tag, là 1 mẫu thông tin nhỏ để xác định message nhận được là hợp lệ tức message nhận được là đúng người gửi và không bị thay đổi.

4. What is the difference between a message authentication code and a one-way hash function?

- Sự khác nhau chính: hash function chỉ đảm bảo tính toàn vẹn của message còn MAC đảm bảo cả tính toàn vẹn và tính xác thực (đúng người gửi).
- + Hashcode được tạo ra từ message mà không dùng bất kì các thông tin nào từ bên ngoài. Nó đảm bảo message không bị thay đổi trong suốt quá trình gửi nhận.
- + MAC thay vì sử dụng private key như hash function, nó sử dụng để tạo đoạn mã. Nó đảm bảo với người nhận rằng, không chỉ message không bị sửa đổi, nó còn đảm bảo người gửi là đúng người.

5. What changes in HMAC are required in order to replace one underlying hash function with another?

- HMAC tạo ra message có tính xác thực có dùng hash functions. HMAC cũng có thể kết hợp với nhiều giải thuật mã hoá khác nhau như MD5, SHA-1... với secret key đã được 2 bên chia sẻ. Độ dài của HMAC phụ thuộc vào hash functions.

Tasks.

Task 1. Generating Message Digest and MAC

```

[vuongMB:Desktop apple$ openssl dgst -md5 "huhu.txt"
MD5(huhu.txt)= 9a8dae267c8e52a664c8cbf7cfe47b44
[vuongMB:Desktop apple$ openssl dgst -sha1 "huhu.txt"
SHA1(huhu.txt)= 5551512d6278c50553c182464f11feed53e8e735
[vuongMB:Desktop apple$ openssl dgst -sha256 "huhu.txt"
SHA256(huhu.txt)= af285d1bccb177d7b89f45e42ffeda0a0c48365576ba1001c9ea104fc7f2a485

```

- Theo quan sát:

- + Điểm giống: cả 3 giải thuật (md5, sha1, sha256) đều mã hoá ra kiểu hex.
- + Điểm khác: cả 3 giải thuật (md5, sha1, sha256) đều tạo ra đoạn mã hoá có độ dài khác nhau mặc dù giải mã cùng 1 file. Đặc biệt độ dài mã phụ thuộc vào giải thuật mã hoá (VD: sha256 tạo 256 bits tương ứng 32 bytes).

Task 2. Keyed Hash and HMAC

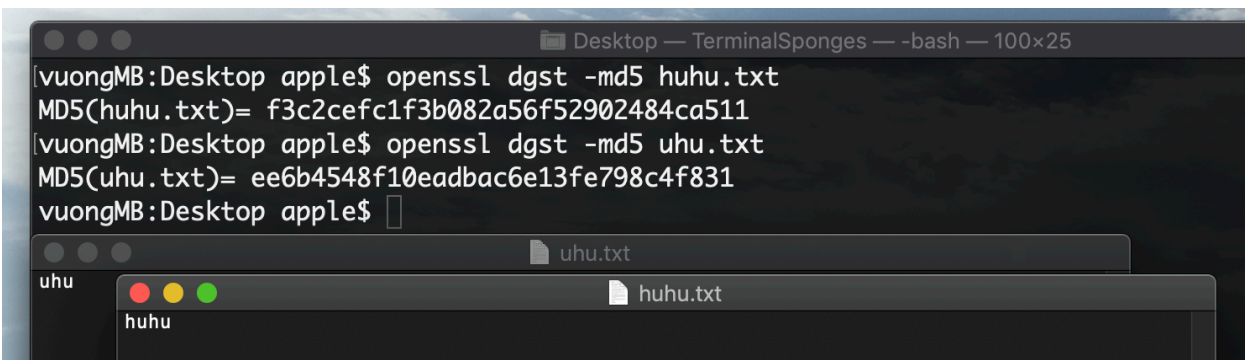
```

[vuongMB:Desktop apple$ openssl dgst -md5 -hmac "huhu" huhu.txt
HMAC-MD5(huhu.txt)= 2d307d9ee22d002bbbd1398aa920c6d2
[vuongMB:Desktop apple$ openssl dgst -md5 -hmac "huhuhaha" huhu.txt
HMAC-MD5(huhu.txt)= 737df8832512ec81f2c4ac6f5f44411d
[vuongMB:Desktop apple$ openssl dgst -md5 -hmac "huhuhahalala" huhu.txt
HMAC-MD5(huhu.txt)= 921bbccf7d32b84458841c29e3cbc6f8
[vuongMB:Desktop apple$ openssl dgst -sha1 -hmac "huhuhahalala" huhu.txt
HMAC-SHA1(huhu.txt)= fcfbd0db9493c11ac7f36bac2f2e211fc8a6b625
[vuongMB:Desktop apple$ openssl dgst -sha1 -hmac "huhuhaha" huhu.txt
HMAC-SHA1(huhu.txt)= b008a0fbebceafad902b6f95b3fc13d976681fd0
[vuongMB:Desktop apple$ openssl dgst -sha256 -hmac "huhuhaha" huhu.txt
HMAC-SHA256(huhu.txt)= 0c286e0ed081834de8c6b5522e8e6ab9fd597e08a274041446cd69c8f9905e07
[vuongMB:Desktop apple$ openssl dgst -sha256 -hmac "huhu" huhu.txt
HMAC-SHA256(huhu.txt)= f436ee19736ad2476a7c0492cf3aa5ee69f4c4b843e56e774f82e565e024d293

```

- Key không ảnh hưởng đến độ dài của HMAC. Độ dài của HMAC phụ thuộc vào hash functions (VD: dùng sha256 kết hợp HMAC sẽ tạo ra đoạn mã 256 bits ứng 32 bytes).

Task 3. The Randomness of One-way Hash



```

[vuongMB:Desktop apple$ openssl dgst -md5 huhu.txt
MD5(huhu.txt)= f3c2cefc1f3b082a56f52902484ca511
[vuongMB:Desktop apple$ openssl dgst -md5 uhu.txt
MD5(uhu.txt)= ee6b4548f10eadbac6e13fe798c4f831
vuongMB:Desktop apple$

```

```
function compare(s1, s2) {  
  var length = s1.length;  
  var diff = 0;  
  for (var i = 0; i < length; i++) {  
    if (s1[i] !== s2[i]) {  
      diff++;  
    }  
  }  
  console.log("Different: " + (diff/length) * 100 + "%" );  
}  
compare('f3c2cefc1f3b082a56f52902484ca511','ee6b4548f10eadbac6e1  
3fe798c4f831');
```

Output: có 12,5% đoạn output là giống nhau.

```
vuongMB:Desktop apple$ node app  
Different: 87.5%
```