TD (UV RES, Module A45) -- Security

TELECOM Lille

Q1. [Symmetric cryptography] Decrypt the following message using the given one-time pad key.

(Note: please explain to the students how OTP works and its major security properties: If the key is truly random, as large as or greater than the plaintext, never **reused** in the whole or part, and kept secret, the ciphertext is **unbreakable**). Encode A~Z letters into 0 ~ 25; modulus is 26. Please explain the concept of **mod.**

Encryption
Plaintext: SECRETMESSAGE
OPT Key: CIJTHUUHMLFRU
Ciphertext: UMLKLNGLEDFXY (Plus the code numbers and mod 26)

Decryption
Ciphertext: NZFYVGJYHSAJRBKSI
OTP key: MRBLACWEDONEABXQE
Plaintext: BIENVENUEENFRANCE (Minus the code numbers)

Q2. [Greatest Common Divisor]
1) GCD (18, 27) = 9
$18 = 2^1 3^2$, $27 = 3^3$, so the answer is: $3^2 = 9$
2) GCD (26, 37) = 1
$26 = 2^1 13^1$, $37 = 37^1$, so the answer is 1
3) GCD (92,108) = 4
$92 = 2^2 23^1$, $108 = 2^2 3^3$, so the answer is 4
4) GCD (232,320) = 8
$232 = 2^3 29^1$, $320 = 2^6 5^1$, so the answer is $8=2^3$

5) GCD (9081, 3270) = 3

Please use the simple prime factorizations (or root finding) for the first three questions, then derive the Euclid's algorithm and apply it to the other two questions. You may find a detailed explanation about this algorithm at Wikipedia. Simply speaking, GCD(a,b) = GCD (b, a mod b) (if a>b); if a < b, then GCD (a,b)=GCD(b,a)=GCD(a,b mod a)

Suppose we have two large integers, say 9081 and 3270. How do we find their greatest common divisor? We do division with remainder:

(1) 9081 = 3 · 3270 − 729..

Now gcd(9081, 3270) = gcd(3270, 729). We have simplified our problem!

Now we can play this game again:

(2) 3270 = 4 · 729 + 354.

1so gcd(3270, 729) = gcd(729, 354).

(3) 729 = 2 · 354 + 21

so gcd(729, 354) = gcd(354, 21).

(4) 354 = 17 · 21 − 3

so gcd(354, 21) = gcd(21, 3) = 3 because 21 = 7· 3. We have found that gcd(9081, 3270) = 3.

This method of computing the gcd is called **Euclid's algorithm**.

The standard Euclidean algorithm to find the greatest common divisor of two positive integers *a* and *b*:

1. Set the value of the variable *c* = max {*a*, *b*}, and set d = min {a, *b*}.
2. Find the remainder *r* when *c* is divided by *d*.
3. If r = 0, then gcd(a, b) = d. Stop.
4. Otherwise, use the current values of *d* and *r* as the new values of *c* and *d*, respectively, and go back to step 2.

Q3. [Modular arithmetic] Which of the following statements are true? Explain your answer.

a) 33 ≡ 333 (mod 11) False; the remainder when 33 is divided by 11 is 0, but the remainder when 333 is divided by 11 is 3.
b) 33 ≡ 3333 (mod 11) True; when either number is divided by 11, the remainder is 0.
c) Any two multiples of 6 are equivalent to each other (mod 3). True; if a number is a multiple of 6, then it is also a multiple of 3, and so the remainder when it is divided by 3 is 0.
d) Any two multiples of 3 are equivalent to each other (mod 6). False; 9 and 6 are multiples of 3, but 9 ≡ 3 (mod 6) and 6 ≡ 0 (mod 6).
e) If *k* is any counting number greater than 1, then any two multiples of *k* are equivalent to 0 (mod *k*). True; if a multiple of k is divided by k, the remainder is always 0.
f) Any two numbers that are 3 more than a multiple of 12 are equivalent (mod 12). True; if a number is 3 more than a multiple of 12, the remainder when that number is divided by 12 is 3. Examples of numbers that are 3 more than a multiple of 12 are 15, 27, 39, etc.

Q4. [Asymmetric cryptography]. Write the pseudocode of RSA algorithm and perform encryption and decryption using the RSA algorithm for the following,
a) p = 7; q = 11, e = 17; M = 8
b) p = 5; q = 11, e = 3; M =9

RSA Algorithm (Outlined as follows)

Step 1. Take two large primes, *p* and *q*, and compute *n*=*p***q* (modulus)

Step 2. Chose a number, *e*<*n,* where *e* is relatively prime to (*p*-1)(*q*-1), i.e., GCD(e, φ(n)) = 1

Choose an integer *e* such that 1 < *e* < φ(*n*) and gcd(*e*, *n*) = 1; i.e., *e* and *n* are coprime.

- *e* is released as the public key exponent.
- *e* having a short bit-length and small Hamming weight results in more efficient encryption – most commonly $2^{16} + 1 = 65,537$. However, much smaller values of *e* (such as 3) have been shown to be less secure in some settings.

Step 3. Find another number *d,* such that (*ed*-1) is divisible by (*p*-1)(*q*-1) ($0<=d<=n$), i.e. $ed \equiv 1 \bmod \varphi(n)$

- This is often computed using the ***extended Euclidean algorithm***. Using the pseudocode in the *Modular integers* section, inputs *a* and *n* correspond to *e* and *φ(*n*)*, respectively. Where The **Extended Euclidean algorithm** is an extension to the Euclidean algorithm, which computes, besides the greatest common divisor of integers *a* and *b*, the coefficients of Bézout's identity, that is integers *x* and *y* such that $ax + by = \gcd(a, b).$ It allows to compute also, with almost no extra cost, the quotients of *a* and *b* by their greatest common divisor.

1. Set the value of the variable *c* to the larger of the two values *a* and *b*, and set *d* to the smaller of *a* and *b*.
2. Find the quotient and the remainder when *c* is divided by *d*. Call the quotient *q* and the remainder *r*. Use the division algorithm and expressions for previous remainders to write an expression for r in terms of *a* and *b*.
3. If r = 0, then gcd(a, b) = d. The expression for the previous value of *r* gives an expression for *gcd(a, b)* in terms of *a* and *b*. Stop.
4. Otherwise, use the current values of *d* and *r* as the new values of *c* and *d*, respectively, and go back to step 2.

Step 4. The public key is the pair (*n*,*e*), the private key is (*n*,*d*) (*d* is hard to obtain unless *n* is factored)

Step 5. **Encryption**(message m) : $c \equiv m^e \bmod n,$

Step 6. **Decryption***:* $m \equiv c^d \bmod n \ (m < n)$

Step 7. **Digital signature**: $s \equiv m^d \bmod n$ (signature), $m \equiv s^e \bmod n$ (verification)

(Note: the key point is Euler's Totient Function, i.e. $\mathbf{a}^{\varphi(n)} \bmod n = 1$, where $\varphi(n) = (p-1)(q-1)$, and gcd (a, n)=1, p and q are primes)

Calculations

a) n = **p*q = 77,** φ(77) = (p-1)*(q-1) = 60, finding e*d = 17*53 = 901 mod 60 = 1, so according to Cipher = $(\text{Msg})^e$ mod N, we got Cipher = 8^{17} mod 77 = 57; Msg = $(\text{Cipher})^d$ mod N = 57^{53} mod 77 = 8

b) n = **5*11 = 55,** φ(55) = (p-1)*(q-1) = 40, finding e*d = 3* 27 = 81 mod 40 = 1, so according to Cipher = $(\text{Msg})^e$ mod N, we got Cipher = 9^3 mod 55 = 14; Msg = 14^{27} mod 55 = 9

Reference: http://www.muppetlabs.com/~breadbox/txt/rsa.html#2

c) Compare RSA with the cryptosystem of **Q1**, identify its advantages and limitations.

1. Security: brute force exhaustive search attack is always theoretically possible for both of the two schemes.

2. Security relies on a large enough difference in difficulty between easy (en/decrypt) and hard (cryptanalysis) problems

3. Speed -- Public key schemes are generally slower because they require the use of very large numbers

4. Public-key cryptography is not necessary in a single-user environment

d) Suppose Alice intends to send two messages to Bob and Clark respectively, and she does not want anybody else to know the content of messages even they are intercepted. What cryptosystem she should choose to use, and explain how does it work. In addition, how Bob and Clark can verify that the messages are really sent by Alice ?

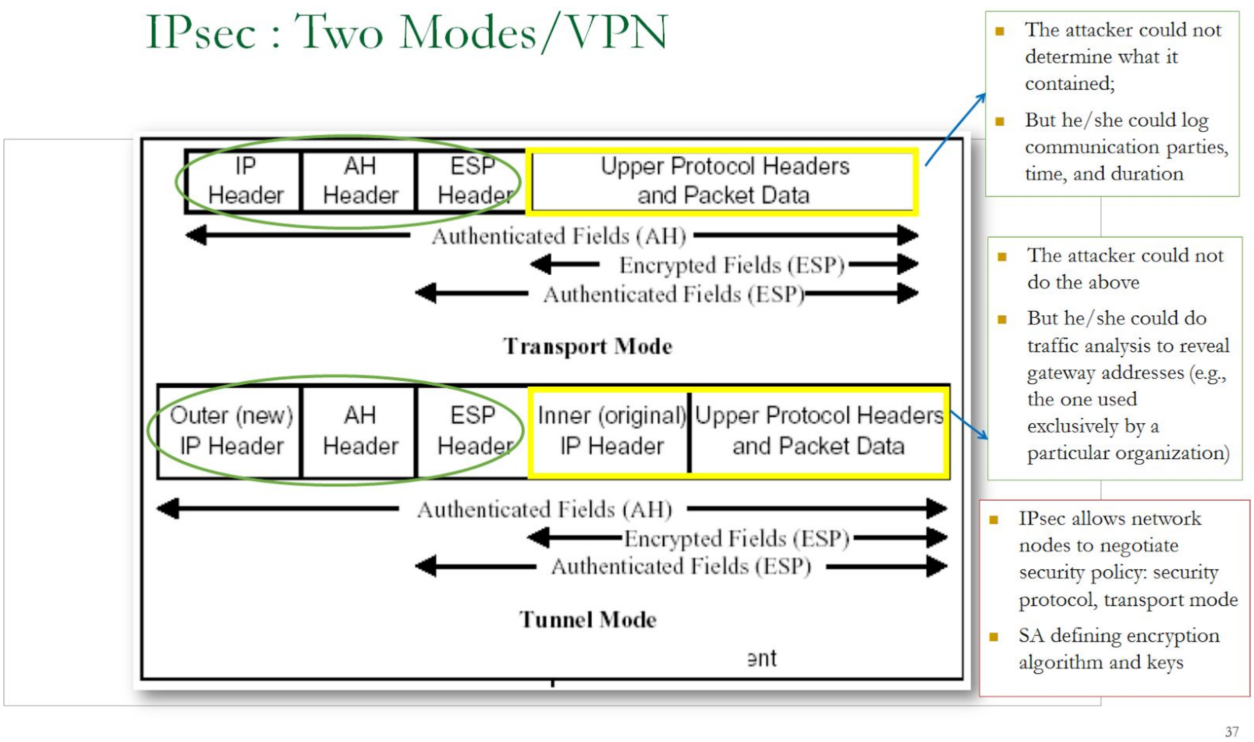
Apparently, we should use asymmetric cryptosystem such as RSA. Then Alice uses the public keys of Bob (K^{+}_{Bob}) and Clark (K^{+}_{Clark}) to encrypt the messages that will be sent to them. When Bob and Clark receive the messages, they use their own private keys (K^{-}_{Bob}, and K^{-}_{Clark}) to decrypt the messages. Two pairs of keys (K^{+}_{Bob}, K^{-}_{Bob}) and (K^{+}_{Clark}, K^{-}_{Clark}) should be generated beforehand.

In order to prove the authenticity of the messages (which are really sent by Alice), digital signature of Alice should be embedded into the messages. To do that, Alice uses her private key (K^{-}_{Alice}) to sign the message (**works for RSA only, other digital signature algorithms such as DSA are available**). On the reception of the messages, Bob and Clark will decrypt the message using Alice's public key (K^{+}_{Alice}).

Q5. [Cryptographic protocols] Suppose your company has multiple branch offices which are geographically located in different cities and even countries, and you are required to choose one of the following protocols to enable site-to-site secure data exchange and communication: (1) SSL/TLS; (2) IPsec; (3) PGP; (4) HTTPs; (5) digest authentication. Explain your reason for choosing this protocol, and analyze its potential attacks.

In fact Virtual Private Network (VPN) is the feasible solution. In addition to other design variants, IPsec is the core technology to enable secure communications between different sites. IPsec is a widely used protocol for securing traffic on IP networks, including the Internet. IPSec can encrypt data between various devices, including router to router, firewall to router, desktop to router, and desktop to server. IPSec consists of two sub-protocols which provide the instructions a VPN needs to secure its packets: **Encapsulated Security Payload** (**ESP**) encrypts the packet's payload (the data it's transporting) with a symmetric key. **Authentication Header** (**AH**) uses a hashing operation on the packet header to help hide certain packet information (like the sender's identity) until it gets to its destination. Networked devices can use IPSec in one of two encryption modes. In **transport mode**, devices encrypt the data traveling between them. In **tunnel mode**, the devices build a virtual tunnel between two networks. As you might guess, VPNs use IPSec in tunnel mode with IPSec ESP and IPSec AH working together.

Attacks to two deployment modes respectively (the blocks on right side)



IPsec : Two Modes/VPN

IP Header
AH Header
ESP Header
Upper Protocol Headers and Packet Data
Authenticated Fields (AH)
Encrypted Fields (ESP)
Authenticated Fields (ESP)
Transport Mode
Outer (new) IP Header
AH Header
ESP Header
Inner (original) IP Header
Upper Protocol Headers and Packet Data
Authenticated Fields (AH)
Encrypted Fields (ESP)
Authenticated Fields (ESP)
Tunnel Mode
ent

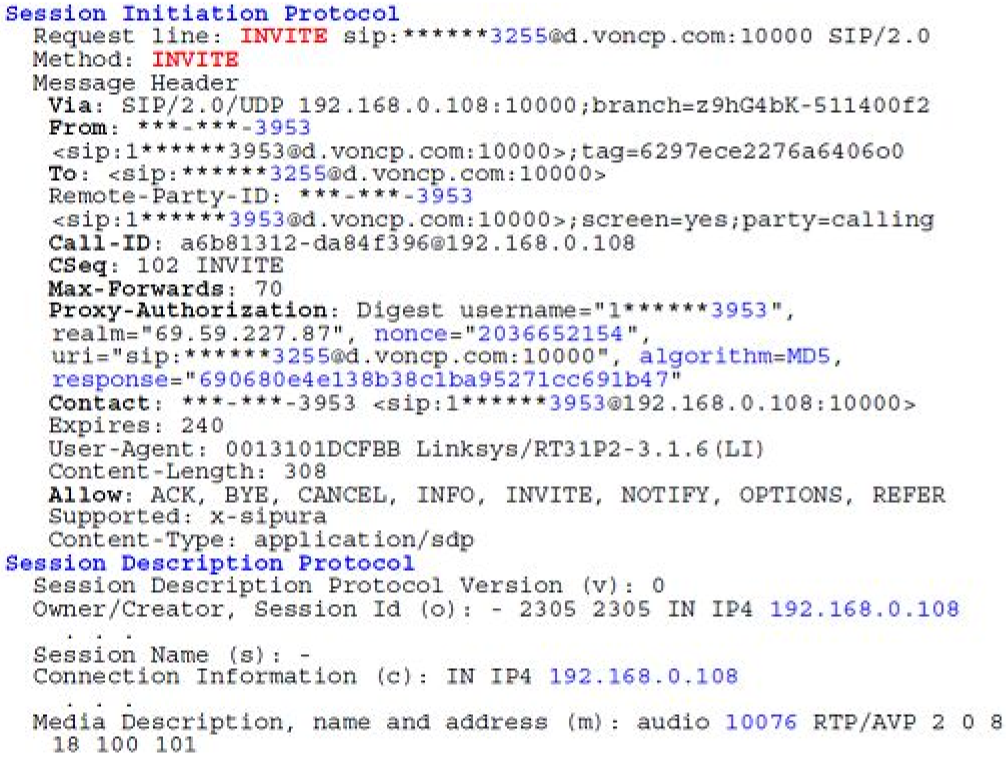
- The attacker could not determine what it contained;
- But he/she could log communication parties, time, and duration

- The attacker could not do the above
- But he/she could do traffic analysis to reveal gateway addresses (e.g., the one used exclusively by a particular organization)

- IPsec allows network nodes to negotiate security policy: security protocol, transport mode
- SA defining encryption algorithm and keys

37

Q6. [SIP Security] Analyze the following message and answer the given questions.

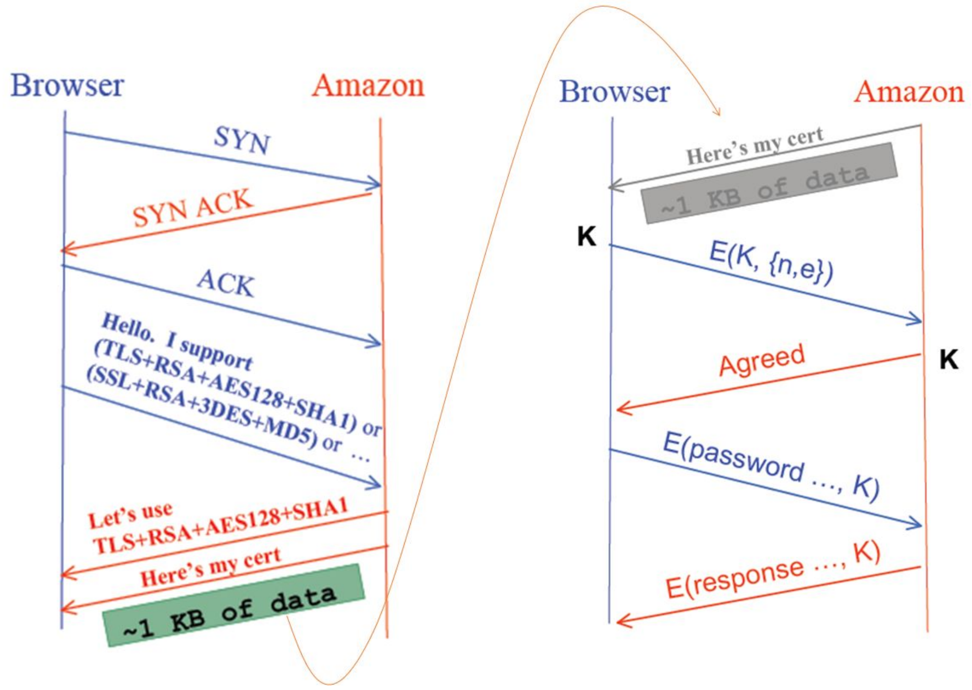


```
Session Initiation Protocol
  Request line: INVITE sip:******3255@d.voncp.com:10000 SIP/2.0
  Method: INVITE
  Message Header
   Via: SIP/2.0/UDP 192.168.0.108:10000;branch=z9hG4bK-511400f2
   From: ***-***-3953
   <sip:1******3953@d.voncp.com:10000>;tag=6297ece2276a6406o0
   To: <sip:******3255@d.voncp.com:10000>
   Remote-Party-ID: ***-***-3953
   <sip:1******3953@d.voncp.com:10000>;screen=yes;party=calling
   Call-ID: a6b81312-da84f396@192.168.0.108
   CSeq: 102 INVITE
   Max-Forwards: 70
   Proxy-Authorization: Digest username="1******3953",
   realm="69.59.227.87", nonce="2036652154",
   uri="sip:******3255@d.voncp.com:10000", algorithm=MD5,
   response="690680e4e138b38c1ba95271cc691b47"
   Contact: ***-***-3953 <sip:1******3953@192.168.0.108:10000>
   Expires: 240
   User-Agent: 0013101DCFBB Linksys/RT31P2-3.1.6(LI)
   Content-Length: 308
   Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, REFER
   Supported: x-sipura
   Content-Type: application/sdp
Session Description Protocol
  Session Description Protocol Version (v): 0
  Owner/Creator, Session Id (o): - 2305 2305 IN IP4 192.168.0.108
    . . .
  Session Name (s): -
  Connection Information (c): IN IP4 192.168.0.108
    . . .
  Media Description, name and address (m): audio 10076 RTP/AVP 2 0 8
   18 100 101
```

a) What security mechanism is applied here, and what security properties can be preserved by using such a mechanism. Is it possible to replace the security mechanism by SSL/TLS, why ?

HTTP digest authentication; authenticity;

They are playing for different goals. HTTP digest authentication is primarily used for preserving the calls' authenticity, while SSLor TLS is used for secure communication between two sides by mainly preserving confidentiality, integrity and anti-replay. *(***note***: please simply explain the handshakes between the two parties and illustrate how the cipher suites are exchanged)*



b) In which call flow context the message will be generated. In particular, what cryptographic hash function is used here; how the RESPONSE value is generated; and what purpose of using NONCE.

MD5; This message is generated in response to the request of SIP server “407 Proxy Authentication Required”; Response=F(nonce, username, password, realm, SIP-method, request-URI). The use of NONCE is to prevent replay attacks.

c) Which of the following attacks could be prevented by using such a security mechanism: (1) SIP registration spoofing; (2) Invite message replay; (3) Fake Bye, and (4) Fake Busy.

Suppose HTTP digest authentication is well implemented, it can prevent Invite message replay. Implementation errors may still lead to attacks. In particular, it Provides **one-way authentication**: identify UAC to a UAS or SIP proxy (but does not identify UAS or SIP proxy !). Also, it must be supported by all SIP compliant UA and SIP servers.

It helpless for other attacks (1) (3) and (4).

For attack (1), The SIP digest does NOT cover the **IP address** of the SIP phone, so the attacker could simply replay the legitimate REGISTER message from different IP address;

For attack (3), basically Attacker could terminate any established SIP call by sending fake BYE message to the SIP phone(s) and/or SIP proxy. While SIP proxy usually requires **digest authentication**, so it could **detect** fake BYE and ignore it, some existing SIP phones however will honor **any** BYE message with **correct Call-ID**;

For (4), **BUSY** message is not protected by SIP authentication.

Appendix: Review the following terms

Symmetric/Asymmetric cryptography	RSA/Diffie-Hellman Key Exchange	DES/AES
Stream cipher/block cipher	Cryptographic hash function/MD5/SHA-1	Digital signature
Public key/Private key	Message Authentication Code	Public Key Infrastructure/Public certificates
HTTPS, TLS/SSL, IPsec, S/MIME, PGP, VPN	VoIP authenticity	HTTP digest authentication
VoIP billing attack	SPam over Internet Telephony(SPIT)	VoIP call hijacking/tampering
Voice phishing (Vishing)	Intrusion Detection System (IDS)	SIP-aware firewall