**Final Exam CMPE-552 17.01.2017 (120 min, 100 points + 5 bonus)**

St. Name, Surname\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ St.Id#\_\_\_\_\_\_\_\_\_\_\_\_\_

**Calculators are allowed. Two A4 sheets of paper with your handwritings may be used for your help (printouts, photocopies, etc. are not allowed). Phones, notebooks, etc., are not allowed.**

**Totally 5 questions, 10 pages. Note that if your point is more than 100, your grade is 100.**

Instructor Alexander Chefranov

**Task 1. (21 points)** HINT:

# “Creating a Blinding Function

The protocol requires that C creates his own blinding function, *b*, unknown to B. This might seem a difficult task, but it is actually quite easy in the context of RSA algorithm for public key cryptography. In one scheme for doing this, C first generates a random number, *u*, that is relatively prime to the modulus *N* of the bank’s keys. Because u is relatively prime to *N*, it has a multiplicative inverse, , with respect to *N*, such that



To blind the serial number, *n*, C computes



and sends the result to B. Hence, the blinding function can be viewed simply as multiplication by a random number.

The signed result, *sr*, returned by B to C is



Obviously, . To recover the token, we use



The serial number *n* can be now obtained using.”

Answer the following sub-questions assuming N=77. Show your calculations. Give necessary explanations.

1. (3 points) Define a redundancy predicate, p(n), for checking validity of a serial number, n

Let p(n)=(nmod3==1)

1. (3 points) Generate a valid serial number, n, and check its validity

Let n=4, p(4)=(4 mod 3==1)=TRUE since 4 mod 3 =1

1. (3 points) Using RSA, define a bank’s private and public keys, P and R,

N=77=pq => p=11, q=7, fi(N)=(p-1)(q-1)=10\*6=60

P\*R=1 mod Fi(N), P\*R=1 mod 60. Take P=7, gcd(60,7)=1, hence R exists. Find it.

A=(1,0,60), B=(0,1,7)

Q=[60/7]=8

T=A-QB=(1,-8,4)

A=(0,1,7), B=(1,-8,4)

Q=[7/4]=1

T=A-QB=(-1,9,3)

A=(1,-8,4), B=(-1,9,3)

Q=[4/3]=1

T=A-QB=(2,-17,1)

A=(-1,9,3), B=(2,-17,1)

B3=1 => 7-1=-17 mod 60 =43

Hence, R=43

Check it: 7\*43=301 mod 60 =(5\*60+1) mod 60 =1

1. (3 points) Blind the serial number, n

Select u=5, gcd(u,N)=gcd(5,77)=1, hence, find u-1

A=(1,0,77), B=(0,1,5)

Q=[77/5]=15

T=A-QB=(1,-15,2)

A=(0,1,5), B=(1,-15,2)

Q=[5/2]=2

T=A-QB=(-2,31,1)

A=(1,-15,2), B=(-2,31,1)

B3=1 => u-1=31

Check it: 5\*31=155 mod 77=(2\*77+1) mod 77 =1

Now, blind n=4:

B(4)=RSAP(u)\*n=57\*4mod77=34

1. (3 points) Get blinded token (signed by the bank blinded serial number), Token

Bank signs B(4):

Token=RSAR(B(4))=3443mod77=34

1. (3 points) Un-blind Token (signed by the bank serial number)

User, on getting Token, un-blinds it:

SSer=u-1\*BToken modN = 31\*34 mod 77=1054 mod 77 = 53

1. (3 points) Get back the serial number, n.

Serial number, n=4, is restored by decrypting SSer with P:

n’=RSAP(Token)=537mod77=4=n

**Task 2. (21 points)** What digital signature and digest methods are used in the XML code below (11 points)? How authenticity of the XML code is expected to be checked (10 points)? Give necessary explanations.

<Signature Id = “A Simple Signature”

 xmlns= “[http://www.w3.org/2000/09/xmldsig#](http://www.w3.org/2000/09/xmldsig) ”>

 <SignedInfo>

 <CanonicalizationMethod

 Algorithm =

 “ <http://www.w3.org/TR/2001/REC-xml-c14n-20010315> ” />

 <SignatureMethod

 Algorithm =

 “[http://www.w3.org/2000/09/xmldsig#rsa-sha256”/](http://www.w3.org/2000/09/xmldsig#rsa-sha256)>

 <Reference URI=”#MsgBody”>

 <Transforms>…</Transforms>

 <DigestMethod

 Algorithm =

 “[http://www.w3.org/2000/09/xmldsig#sha256”/](http://www.w3.org/2000/09/xmldsig#sha256)>

 <DigestValue>dER4boXp453tr56Y</DigestValue>

 </Reference>

 </SignedInfo>

 <SignatureValue>zi990CrnT9zopr0o</SignatureValue>

</Signature>

Digital signature method is RSA-SHA256 as shown in SignatureMethod element’s attribute Algorithm

Digest method used is SHA256 as shown in the DigestMethod aelement’s attribute Agorithm

Authenticity of the XML code is validated by comparing SHA256(XML) versus result of RSA decryption of the SignatureValue specified. If they comply, the code is considered authentic.

Authenticity of the #MsgBody is checked by calculating SHA256 on it and comparing versus DigestValue in the XML code

**Task 3. (21 points)** Kerberos with multiple realms is described as follows:



A user wishing service on a server in another realm needs a ticket for that server. The details of the exchange illustrated in Figure 14.2 are as follows:

1. C -> AS: IDc||IDtgs||TS1
2. AS -> C: EKc[Kc,tgs||IDtgs||TS2||Lifetime2||Tickettgs]
3. C -> TGS: IDtgsrem||Tickettgs||Authenticatorc
4. TGS -> C: EKc,tgs[Kc,tgsrem||IDtgsrem||TS4||Lifetime2||Tickettgsrem]
5. C -> TGSrem: IDvrem|| Tickettgsrem|| Authenticatorc
6. TGSrem -> C: EKc,tgsrem[Kc,vrem||IDvrem||TS6|| Ticketvrem]
7. C -> Vrem: Ticketvrem|| Authenticatorc

Explain, how using Multiple Kerberi, a user, A, can get medical statistics for year 2016 from a server, Med\_stat, located in Turkey. Assume that A is registered with AS\_TRNC and TGS\_TRNC, Med\_stat is registered with AS\_Turkey and TGS\_Turkey. Show **all 7** necessary steps of the Multiple Kerberi protocol, give explanations on the messages exchanged (from whom, to whom, content of the messages, particular values). Assume that the first step of the protocol is implemented on 17.01.2017, 16:30:29.

1. A -> AS\_TRNC: IDA||IDtgs\_TRNC||17.01.2017, 16:30:29
2. AS\_TRNC -> A: EKc[Kc,tgs\_TRNC||IDtgs\_TRNC||TS:17.01.2017, 16:31:29|LT: from 17.01.2017, 16:31:29 to 17.01.2017, 18:31:29||Tickettgs\_TRNC]
3. A -> TGS\_TRNC: Idtgs\_Turkey||Tickettgs\_TRNC||Authenticator1A
4. TGS\_TRNC -> A: EKA,tgs\_TRNC [KA,tgs\_Turkey || IDtgs\_Turkey || TS: 17.01.2017, 16:32:29|| LT: from 17.01.2017, 16:32:29 to 17.01.2017, 18:32:29|| Tickettgs\_Turkey]
5. A -> TGS\_Turkey: IDMed\_stat|| Tickettgs\_Turkey|| Authenticator2A
6. TGS\_Turkey -> A: EKA,tgs\_Turkey[KA,Med\_stat||IDMed\_stat|| TS: 17.01.2017, 16:33:29|| TicketMed\_stat]
7. A -> Med\_stat: TicketMed\_stat|| Authenticator3A || Statistics(2016)

**Task 4. (21 points)** Consider X.509 strong authentication procedures below



Who is authenticated to whom and how it is made in Step 2 of the two-way authentication? What information and in what form is used for that authentication? Give necessary explanations.

B is authenticated to A in Step 2 of the two-way authentication

For the authentication, B provides identifier of A, IDA, A’s nonce, rA, digitally signed with B’s private key. IDA is used to show that B wants being authenticated by A, rA, is returned to A confirming that B has authenticated A on getting A’s first message. This information is not hidden as conveyed just digitally signed. Timestamp, tB, and nonce, rB, are conveyed to counter replay attack in authentication of B to A.

**Task 5. (21 points)** Consider Lamport’s One-time password scheme below.

# Initialization Procedure

The client selects a password, , a number, , calculates

,

where

.

The client securely delivers to the server (, and the servers saves it into () tuple.

# Authentication Procedure

When the client, C, requests authentication by the server, S, the following proceeds:

1. C -> S: C\_ID //client sends his ID

2. S -> C: Counter(C\_ID) //server responds by respective Counter value

3. C -> S: C\_ID, 

4. S: If  then {

 S authenticates C, and sets ()=()

 }

 Else C is not authenticated

Assume that the client, A, specified N=7. Specify all the rest Lamport’s OTP scheme ingredients necessary to specify and show record of A in the server, S, after the second successful authentication of A. Show your definitions, intermediate calculations, and give necessary explanations

Let p0=1, h(x)=(2x+1) mod 11.

Then

H(p0)=3; h2(p0)=h(3)=7; h3(p0)=h(7)=4; h4(p0)=h(4)=9; h5(p0)=h(9)=8; h6(p0)=h(8)=6; h7(p0)=h(6)=2;

Pwd7=2;

Initial A’s record is (7,2,A)

After the first authentication, his record is (6,6,A)

After the second authentication, his record is (5,8,A)