Ministry of Education, Culture and Research of the Republic of Moldova
Technical University of Moldova
Department of Software and Automation Engineering

REPORT

Laboratory work no. 5
Public key cryptography

Elaborated:
st. gr. FAF-213 Botnari Ciprian

Verified:
asist. univ. Cătălin Mîțu

Chişinău – 2023

Table of Contents

Topic: Public keys

Tasks

1. Generate keys, encrypt and decrypt the message: <Last name> <First name> with RSA. The value of n must be at least 2048 bits (617 digits).
2. Generate keys, encrypt and decrypt the message: <Last name> <First name> with ElGamal. The values of prime and generator are given below.
3. Use the Diffie-Helman key exchange between Bob and Alice, which has to be prepared for AES 256 bits

Theoretical notes

RSA, or Rivest-Shamir-Adleman, is a widely used asymmetric-key cryptosystem that offers secure communication and digital signatures. The RSA algorithm relies on the mathematical properties of large prime numbers. Here are the key phases of the RSA algorithm:

1. Key Generation: RSA begins with key generation, which involves selecting two large prime numbers, p and q. The product of these primes, n = p * q, becomes the modulus used for encryption and decryption. The public key (e, n) and private key (d, n) are generated, with e as the encryption exponent and d as the decryption exponent.
2. Encryption: To encrypt a message M, the sender uses the recipient's public key (e, n). The message is raised to the power of e and then taken modulo n to obtain the ciphertext $C = M^e \bmod n$.
3. Decryption: The recipient uses their private key (d, n) to decrypt the ciphertext C. The ciphertext is raised to the power of d and taken modulo n to recover the original message: $M = C^d \bmod n$.
4. Digital Signatures: RSA is also used for digital signatures. To sign a message, the sender applies their private key to a hash of the message, creating a signature. Recipients can verify the signature using the sender's public key.
5. Key Security: The security of RSA relies on the difficulty of factoring the modulus n, given that it is a product of two large prime numbers. The prime factorization problem forms the basis of RSA's security.

ElGamal is another asymmetric-key cryptosystem, primarily used for secure key exchange and digital signatures. It involves the following phases:

1. Key Generation: ElGamal starts with key generation. The sender selects a large prime number, p, and a primitive root modulo p, g. The sender also generates a private key, x, which is a random number, and computes the corresponding public key, $y = g^x \bmod p$.
2. Key Exchange: In the key exchange phase, two parties can securely exchange a shared secret by using their respective private keys and the other party's public key. This shared secret can be used as a symmetric key for encryption and decryption.
3. Encryption: ElGamal encryption is probabilistic and involves generating a random number, k. The message is encrypted as a pair of ciphertexts (c1, c2), where $c1 = g^k \bmod p$ and $c2 = (M * y^k) \bmod p$, with M being the plaintext message.
4. Decryption: The recipient uses their private key, x, to decrypt the ciphertext (c1, c2). They calculate the shared secret, S = c1^x mod p, and then compute the plaintext as M = (c2 / S) mod p.
5. Digital Signatures: ElGamal can also be used for digital signatures, where the sender signs a message using their private key, and the recipient verifies it using the sender's public key.

Diffie-Hellman is a key exchange protocol that allows two parties to securely exchange a shared secret over an untrusted network. Here are the key phases of the Diffie-Hellman protocol:

1. Parameter Selection: The parties agree on two public parameters, a prime number p and a primitive root modulo p, g.
2. Key Generation: Each party generates its private key, a random number (a for the first party, b for the second party). They then compute their corresponding public key: $A = g^a \bmod p$ for the first party and $B = g^b \bmod p$ for the second party.
3. Key Exchange: The parties exchange their public keys (A and B) over the insecure channel.

4. Shared Secret: Each party uses the received public key to compute the shared secret. The first party computes $S = B^a \bmod p$, and the second party computes $S = A^b \bmod p$. Both parties arrive at the same shared secret, which can be used as a symmetric encryption key.
5. Encryption: Once the shared secret is established, the parties can use it for secure communication, such as encrypting and decrypting messages using a symmetric encryption algorithm.

Results

Here are the results for each one.

RSA

Message to encrypt: Botnari Ciprian
Message (hex): 0x426f746e61726920436970726961e
Message (decimal): 3449521647573127874258009867792797 26

Cipher text:
175652160112928478283778997333251724861722274775457590897433617655424825257509268201336
4295704446473575048000226543224084592591314059079196787616920008360133045729201033476 60
0367130180733678868948763388544785844192349593366729592721806604124056711568015446030 89
228980622747004389993579595318033270541072991403387069006826048468445854023405551341782
506885274217692491837548458566586627587017423058596349907753962325308367565358937244459
102547964891106458299101541893331970650129545721957645724563478034549012041889129475611
457712068088696960998660015883622553412485057505718714698504331167487629564957555977673
604660485744744411540551154535968811872882459452570529828452731228864479951157053562668
172290040199713835712376844769856652969684638591801539010225205712191361249207379861432
012213338144324551359681260001461057482880406204429493701382132350035682425375651557950
454006539498655985888448443809988911410876845764763458290211446127590711220856142659822
786248772716725339501659693298895172443202054534453093042875954333086780434271338397698
240067888349066319741883794408642448361536697043928322253062234324635025365624452090580
128526462392204910650276279145580670912466237729932597904081561452783020028792952726706
052917229588978658873554533918866142708263652228654181121417279224622765678874915271998
859506083064622179533525172445625185456362625765497716377891205592472222658093103157128
887706324418125824375472368285395423265090627297870370060414366139790793836142392438235
445246138126309999705692761505315510496561669788391904265019531300178172747413828798594
891695263616195516699111592140898895901558210064311206935509355164256546451567347209923
332212619835918771651822219978026328895751715529454495022454068308446122061456751575660
071471706821472907546663616796266372764762802881892844810396033794438355527890601288067
371189420799854904493907115624320551207261853597349452035170489337741790561189424092478
499227307070251276476553624694699983302200522374839597958141131340739589790567463783311
840088757155213512469441445761383663804176870449534748752325176313845220306760936871372
419147496333432928456844480286474818979506726918191908575288196052442665368025663445282
378476999226653130351045191446910737563043945503379866351387468269461048469115944943477
284594250707818543165813105798365744842946316466456984582881328694501081933828795833481
956825481496577108420249741986649899325270200039773640802337434181397241211611143201736
440579209180456713868334678253301618048295391675 8386

Decrypted cipher: Botnari Ciprian

ElGamal

Message to encrypt: Botnari Ciprian
Message (hex): 0x426f746e617269204369707269616e
Message (decimal): 344952164757312787425800986779279726

Cipher text:
315873841690284449372388648748692761894122605912688839117497715187048520032582179471713
114485316274309552730117866345452698588056993700118477611238100395232868388978244070952
940860133789852778083356451969712446973828673488599231244838241058018958344489377054243
763160070490848360954282660447600552496250883187917791412396507424633436210899084788103
378652133360974290700114499249603172520063830508681748258641781008408820253807624722860
010614723022718231047680248766964123848149244624697293553896707613656154343889589589771
332850891890389751426101130401704146981638591094274232833238233384346605612298019838283
7154851

Decrypted cipher: Botnari Ciprian

Diffie-Hellman

Prime:
323170060713110073001535134778251633624880571334890751745884341392698068341362100027920
563626401646854585563579353308169288290230805734726252735547424612457410262025279165729
728627063003252634282131457669314142236542209411113486299916574782680342305530863490506
355577122191878903327295696961297438562417412362372251973464026918557977679768230146253
979330580152268587307611975324364674758554607150438968449403661304976978128542959586595
975670512838521327844685229255045682728791137200989318739591433741758378260002780349731
985520606075332341226032546840881200311059074842810039949669561196969562486290323380728
39127039
Generator: 2

Private key 1:
171767727904857686930604581903021808576012201467695207496406203675122244441982303111582
671769890595008178209361842542879699792075437054391829391703255160719003168755112029148
438518714781543541811949826890753780792779144215099784032384507682335705379947230853641
954791650046668011760754510519168180708544225181915989684120112189165847534316833054348
251470703925744935345197086710611919541152642747174038825305868614320395769787574778620
606160504434372423216035329374863541021959566797070036105664894096129425710336594396182
436289186980148191730860851767078751477971103102970176974637952216064669612824298124073
73256890
Private key 2:
697637892012988336881029036614137988128003296616618416643342797064868434394433002166334
438857841053783343597661119954212780798298208459838836850404559397664199313351673614387
121660851526392663401122671528491445571140241535389283476042146659999445677954416546354
577699980058277158573265241226192841874822864495423679234296779084337643760844808163893
226708709135126148171278511528181996692795229337359984749034200052021458073609538013361
267226849065732817614849003031994564079237297511211795895009601115290303688661482097224
511896906790065378623457877293847382317794924291458819563380905045345856230279850087308
9485750

Public key 1:
2358891421901785953288658216456352754669442566932935674152032401834042022723259000732604170708897596423406545351762061291443874023808171354478580008467824208269936052984639288428568264142380743956978067629720738999805283183155024136843544865125078719398115491021572109878833768504092207112777449793097871596275408780387230269279626494367003345844596678676180520091509677725897339835863195424354729467725470279682710854170874632736922653816496578971200418146159589476298102857769858409928458665322611922033999592334512547114237404683514138069247603250691568225431228014875692627284261454749035015017384481747277 1731820
Public key 2:
28579068021347094198681757418256654730793728062517197023108114028614810292912562401124842322123617829973170827695536752112153381959280115453300068175303400862682755100134599393526724765079774511458672975073289866212086175876777817802318068060562939695163577098251259382633533998199834722059375434761618452260789063604999558104664172881339280366114143063744827975026765300865217781914884656576683744894825261754747265681769374968021909797042224593208403640995727101349779202730899296931451125709295653900121224455854110040002231613314194713058941771465276342459327481116922504185604174354726102202105227554540979061653

Shared secret 1:
24987627590650235790650448857643353111459168490155178937164262233607947737681525692463786904424162855042542077994074341603840396119449581470012820081983229303277057077293262907961088129053988809261261903770873602222616483137616253337169768241476059534505563012069674248285060054166995268670432966094129742135650601763135996396738386852257549027307107086136757956168175208628039939176226281006310091584378925343455748090917081376813397911341552912799367417884146014710471885737187823952128562484469535148597718449024366668350517544779163078137848233087601337164081094825020757065207293667496800634447716837440863581312
Shared secret 2:
24987627590650235790650448857643353111459168490155178937164262233607947737681525692463786904424162855042542077994074341603840396119449581470012820081983229303277057077293262907961088129053988809261261903770873602222616483137616253337169768241476059534505563012069674248285060054166995268670432966094129742135650601763135996396738386852257549027307107086136757956168175208628039939176226281006310091584378925343455748090917081376813397911341552912799367417884146014710471885737187823952128562484469535148597718449024366668350517544779163078137848233087601337164081094825020757065207293667496800634447716837440863581312

Shared secrets match!
Shared secret (256 bits):
b'\xabQ\xc5\xd2\xe6\xc3\x88\x94\xe6\xa3\xb2\xdbj\x90\xca\x90F\n\x17\xd2\xd6\xff\xad\x95\x83\xc5Pd\xc0;\xb4\x80'
Shared secret hex: ab51c5d2e6c38894e6a3b2db6a90ca90460a17d2d6ffad9583c55064c03bb480
Shared secret decimal:
77489977049906984294413060619135534752564390230583346125826464717709866939520
Shared secret length: 32 bytes, 256 bits, 77 digits.

Conclusion

In conclusion, the study and analysis of RSA, ElGamal, and Diffie-Hellman encryption algorithms have provided valuable insights into the field of modern cryptography. This exploration has equipped me with practical knowledge and experience in the implementation of these algorithms for secure communication and data protection.

RSA, as a widely used public-key cryptosystem, has shown its strength in ensuring data confidentiality and digital signature authenticity. Its reliance on the factorization problem makes it a robust choice for securing communications and digital transactions in an era where cyber threats are on the rise.

ElGamal, another public-key encryption method, has demonstrated its importance, particularly in the context of key exchange and digital signatures. Its security relies on the discrete logarithm problem, which makes it suitable for various cryptographic applications, further contributing to the diversity of cryptographic tools available.

The Diffie-Hellman key exchange protocol, a foundation for secure key establishment, has illustrated its critical role in securing communication channels. By enabling parties to exchange secret keys over an insecure medium, Diffie-Hellman addresses the issue of key distribution and has had a significant impact on the development of secure communication systems.

GitHub

Sufferal/cryptography-labs (github.com)