ФГБОУ ВПО «Чувашский государственный университет им. И. Н. Ульянова»

Факультет ИВТ

Кафедра Вычислительной Техники

Лабораторная работа №2

**Атаки на шифрование методом RSA. закладки**

**Выполнил:**

Студент гр. ИВТ-21-11

Исаков Тимофей

**Проверил:**

Лобастова О. А.

г. Чебоксары, 2015г.

**Цель работы.**

Целью работы является знакомство с классическими атаками на шифрование с использованием алгоритма RSA.

**Выполнение работы.**

1. Неудачный выбор параметров криптосистемы. Параметры p и q близко расположенны друг к другу.

**Test #1**

Key's length is 1024

Difference: 75

p: 12096001276411371097975737433975882263963623773337505199393143170829772661264599006232251172761346851609344383500173475351566391120652495721064529697673123

q: 12096001276411371097975737433975882263963623773337505199393143170829772661264598006232251172761346851609344383500173475351566391120652495721064529697672567

Message: 12096001276411371097975737433975882263963623773337505199393143170829772661264599006232251172761346851609344383500173475351566391120652495721064529697673023

Start factorization

Start test

{'public\_key': {'e': 262145, 'n': 146313246878945518828217308217076239003408835016749753500750981695317344458716617970547213743517230574747151840638052905382870229079654700375357275661129549501847559812203922260221205074273878173311157482753344433923214159850911319300035282662230511604021567162946085140571078690494874122674238777408650316741L}, 'private\_key': {'d': 119286500789494341585779852931659836931036422733410100584361713211721045575562505582411610428167577306056231420339300551390405276085227495750909258879047566836181100474171695321893931459676680284915626958562771861772101945527517394565596302591187852523950706946912977017536607240059735303213117416996316812921L, 'n': 146313246878945518828217308217076239003408835016749753500750981695317344458716617970547213743517230574747151840638052905382870229079654700375357275661129549501847559812203922260221205074273878173311157482753344433923214159850911319300035282662230511604021567162946085140571078690494874122674238777408650316741L}}

Crypt message: 11366552294233492186327375951720901386125552711837889340535462902141694616715398993904931377823792809843912433310695790140346621755856438155613177053487248203212011779211304267021302423365692558247319994984763167141484540640033698018775756369018984638908815392488603181547028761549610340933938972550896306613

Steps: 0

Factorization p: 12096001276411371097975737433975882263963623773337505199393143170829772661264599006232251172761346851609344383500173475351566391120652495721064529697673123

Factorization q: 12096001276411371097975737433975882263963623773337505199393143170829772661264598006232251172761346851609344383500173475351566391120652495721064529697672567

{'public\_key': {'e': 262145, 'n': 146313246878945518828217308217076239003408835016749753500750981695317344458716617970547213743517230574747151840638052905382870229079654700375357275661129549501847559812203922260221205074273878173311157482753344433923214159850911319300035282662230511604021567162946085140571078690494874122674238777408650316741L}, 'private\_key': {'d': 119286500789494341585779852931659836931036422733410100584361713211721045575562505582411610428167577306056231420339300551390405276085227495750909258879047566836181100474171695321893931459676680284915626958562771861772101945527517394565596302591187852523950706946912977017536607240059735303213117416996316812921L, 'n': 146313246878945518828217308217076239003408835016749753500750981695317344458716617970547213743517230574747151840638052905382870229079654700375357275661129549501847559812203922260221205074273878173311157482753344433923214159850911319300035282662230511604021567162946085140571078690494874122674238777408650316741L}}

Decrypt message: 12096001276411371097975737433975882263963623773337505199393143170829772661264599006232251172761346851609344383500173475351566391120652495721064529697673023

End test

Test time: 0.0330171585083

**Test #2**

Key's length is 2048

Difference: 75

p: 92571543272706031700230177849026936649892081865915834583211396472348699033608917806954776349099996444958325366529619168877470303051412012814704287525896514686594866585643013742762888468327885830827556797338251076527174166148638902041333305901733247094673969104596618602447899169708127038806521806128017436533

q: 92571543272706031700230177849026936649892081865915834583211396472348699033608917806954776349099996444958325366529619168877470303051412012814704287525896514686594866585643013742762888468327885830827556797338251076527174166148638902040333305901733247094673969104596618602447899169708127038806521806128017435903

Message: 92571543272706031700230177849026936649892081865915834583211396472348699033608917806954776349099996444958325366529619168877470303051412012814704287525896514686594866585643013742762888468327885830827556797338251076527174166148638902041333305901733247094673969104596618602447899169708127038806521806128017436433

Start factorization

Start test

{'public\_key': {'e': 1048577, 'n': 8569490623890485354163021762984635814074329785861202256053422780577722790579563255485684118918438245988177003311553024416522400926310301915286944857170291295210768770041577363056514066656566354814480462749547540404154802757374488667529716872502067548190599387086573006277958070895927537907760808409728363763267240006750967953394812547398950341407951502339020734504393201554237873006035174765861410332983784148714033719635384976155021420629114580842938377314096462423831057581824252470908283896226219927182282691012909937313493751245169637231971531165004105418858449579144129033269301286811868300450803391829698044299L}, 'private\_key': {'d': 5081951799054875179115426717358204496265629454699832788717696657644911941805736056139125567098807207417580239726069603368252997826972261361630450259926877337428423099960779514650436239926541013022808313585294779815653028749853108102754503504396144180061788080864224765498251759263878567298739482229985825369982289159950048544275268282137142676353296272157543050852252306705298905825756638672117675043590898542696940183544522453798006873488327908862926154321124454946540500703478569215241528850729559457857985367015914397257015889855376635153024536906780429111561571871445571256081780907645980377094425292840648242465L, 'n': 8569490623890485354163021762984635814074329785861202256053422780577722790579563255485684118918438245988177003311553024416522400926310301915286944857170291295210768770041577363056514066656566354814480462749547540404154802757374488667529716872502067548190599387086573006277958070895927537907760808409728363763267240006750967953394812547398950341407951502339020734504393201554237873006035174765861410332983784148714033719635384976155021420629114580842938377314096462423831057581824252470908283896226219927182282691012909937313493751245169637231971531165004105418858449579144129033269301286811868300450803391829698044299L}}

Crypt message: 4888022240576258842101292344573278377715278083181292310126192663695902899672996693494205468012873812144954078499377025970663861799720310324625158463657479232196656914326860579233009756609793576748456170530188540637613786488978978096415318450712975282897984748223341205899553927078067966789978523765419310187092843573841706111740251382824777187422500900438103249530255572482222383653619931611478566919157454449294981346822265377053989555070139795667752975611126074237742394772333168717716320254563380234178308638759614046019651646496209852862522314506310869009299048800448198160394789430559327755144088736988097090893

Steps: 0

Factorization p: 92571543272706031700230177849026936649892081865915834583211396472348699033608917806954776349099996444958325366529619168877470303051412012814704287525896514686594866585643013742762888468327885830827556797338251076527174166148638902041333305901733247094673969104596618602447899169708127038806521806128017436533

Factorization q: 92571543272706031700230177849026936649892081865915834583211396472348699033608917806954776349099996444958325366529619168877470303051412012814704287525896514686594866585643013742762888468327885830827556797338251076527174166148638902040333305901733247094673969104596618602447899169708127038806521806128017435903

{'public\_key': {'e': 1048577, 'n': 8569490623890485354163021762984635814074329785861202256053422780577722790579563255485684118918438245988177003311553024416522400926310301915286944857170291295210768770041577363056514066656566354814480462749547540404154802757374488667529716872502067548190599387086573006277958070895927537907760808409728363763267240006750967953394812547398950341407951502339020734504393201554237873006035174765861410332983784148714033719635384976155021420629114580842938377314096462423831057581824252470908283896226219927182282691012909937313493751245169637231971531165004105418858449579144129033269301286811868300450803391829698044299L}, 'private\_key': {'d': 5081951799054875179115426717358204496265629454699832788717696657644911941805736056139125567098807207417580239726069603368252997826972261361630450259926877337428423099960779514650436239926541013022808313585294779815653028749853108102754503504396144180061788080864224765498251759263878567298739482229985825369982289159950048544275268282137142676353296272157543050852252306705298905825756638672117675043590898542696940183544522453798006873488327908862926154321124454946540500703478569215241528850729559457857985367015914397257015889855376635153024536906780429111561571871445571256081780907645980377094425292840648242465L, 'n': 8569490623890485354163021762984635814074329785861202256053422780577722790579563255485684118918438245988177003311553024416522400926310301915286944857170291295210768770041577363056514066656566354814480462749547540404154802757374488667529716872502067548190599387086573006277958070895927537907760808409728363763267240006750967953394812547398950341407951502339020734504393201554237873006035174765861410332983784148714033719635384976155021420629114580842938377314096462423831057581824252470908283896226219927182282691012909937313493751245169637231971531165004105418858449579144129033269301286811868300450803391829698044299L}}

Decrypt message: 92571543272706031700230177849026936649892081865915834583211396472348699033608917806954776349099996444958325366529619168877470303051412012814704287525896514686594866585643013742762888468327885830827556797338251076527174166148638902041333305901733247094673969104596618602447899169708127038806521806128017436433

End test

Test time: 0.0684130191803

**Test #3**

Key's length is 4096

Difference: 75

p: 17503762332113782658072116958792969590172107004036375735420444736728421220935737658128087004838074801181476533575001826642794290112620108806684664313882807227172485515368818233672315897609546729290853787048415449983071727097955613475151491575093295492157765919755561793829693214857805721056057829950873858879472026454384710689192690656674354278485467349305223941260153694737104606520693221358365673196570008301216264172120082403181589809059764520567384671003874108498191393934858866889078386413566549664890319575518501345745782153763330069795593858074251540603595930632699484101018252108339980564312900420337638193003

q: 17503762332113782658072116958792969590172107004036375735420444736728421220935737658128087004838074801181476533575001826642794290112620108806684664313882807227172485515368818233672315897609546729290853787048415449983071727097955613475151491575093295492157765919755561793829693214857805721056057829950873858879472026454384710689192690656674354278485467349305223941260153694737104606520693221358365673196570008301216264172120082403181589809059764520567384671003874108498191393934858866889078386413566549664890319575518501345745782153763330069794593858074251540603595930632699484101018252108339980564312900420337638192073

Message: 17503762332113782658072116958792969590172107004036375735420444736728421220935737658128087004838074801181476533575001826642794290112620108806684664313882807227172485515368818233672315897609546729290853787048415449983071727097955613475151491575093295492157765919755561793829693214857805721056057829950873858879472026454384710689192690656674354278485467349305223941260153694737104606520693221358365673196570008301216264172120082403181589809059764520567384671003874108498191393934858866889078386413566549664890319575518501345745782153763330069795593858074251540603595930632699484101018252108339980564312900420337638192903

Start factorization

Start test

{'public\_key': {'e': 262145, 'n': 306381695779125327432808059618915510880273889907762024009549947528695792080713367102958856204751601919073984816347958715634835227308556573067575834323990535346915214683971022425626571015468953096289862734958120198053039687953338692474806821888509122865983638547943161556706400172484995711263391082685831264381190047637059582381247891291920624001831887930228974238504619890475241949508460551959634598621522290560411814119868990013728793372272732859285759337443179542122880654691888467415072842370769296314476674638283069630265660246396846732344334584162434155994527154422311698982399031743301968191595349387737251434697273796890206780693046718883975646575118077838974768491557455038215110742060681141644794711568307082453681626994370617813586838002880918138080937982663293752620253352303147201502026748484275664806085252962118968372620880072172769959337138632496618469772890325508831976320450394554348960239250187755180549962162426583533314440044730925579357480014691292943249539323267297815707159720047875747083620681897354797912953604087320288044313662432085892004515863926684921271448430789653321766094770917566238651424712388059207056344946245396899682002113393438044058276872775814429833048618019509490327484361680980327558665219L}, 'private\_key': {'d': 294666156217089231828518571016729663043909032777412764902293548688253873982649048531786186214492661036597501863024141979990349967934771182965039527458463132091779781576369788265926859985851524627933763690325492465823610670302632911119574169796688125144842209213778549409061299349165879966123471449609317225989624124054638078046663486240097349344698000789094810940456706293869837210520981070860077570196100713034319124864923952893441862762252851212939346277501046631976817385575058125438805165428906619451456917684787296336959348936587840298328734020079029017694391198382985171790960828099536651557089439367471069709353545689675126764805889989937672430975520346842056294925633070349434165679726879253477290964854384543050428485138078710117543530294899963977916098579664528418853926087453196253243420904507699948560896926407886067232674646052180802551949113331764811229132711586519781939985947120407054652225795969013085655220994562510271595806132548336169690198656144333555018354402407274794357335881175681996778293486260614786407218512418412280214878011734445233737555028143677128377553739184867460295282824454245144934720194505297736697900778224104150999150993204844327989012893710950108568448701971111936685330540295777868252534865L, 'n': 306381695779125327432808059618915510880273889907762024009549947528695792080713367102958856204751601919073984816347958715634835227308556573067575834323990535346915214683971022425626571015468953096289862734958120198053039687953338692474806821888509122865983638547943161556706400172484995711263391082685831264381190047637059582381247891291920624001831887930228974238504619890475241949508460551959634598621522290560411814119868990013728793372272732859285759337443179542122880654691888467415072842370769296314476674638283069630265660246396846732344334584162434155994527154422311698982399031743301968191595349387737251434697273796890206780693046718883975646575118077838974768491557455038215110742060681141644794711568307082453681626994370617813586838002880918138080937982663293752620253352303147201502026748484275664806085252962118968372620880072172769959337138632496618469772890325508831976320450394554348960239250187755180549962162426583533314440044730925579357480014691292943249539323267297815707159720047875747083620681897354797912953604087320288044313662432085892004515863926684921271448430789653321766094770917566238651424712388059207056344946245396899682002113393438044058276872775814429833048618019509490327484361680980327558665219L}}

Crypt message: 202476381165267652122945110040054773759326603806121339350301678505210242458902976011304143967537643441927800476951728920827900706030002330931029488659048200026336387495869784188594574896632131431054604549045783580181474891805868220092761149451454610432700553451507337392102985972834747605917989724206726409865608906327862419605028185678892345725234418359417748269195338073517517338638157925239630366306486238741109473527505180634555800093743848857131223266684928159815863700488323977207502856224914640923982399798680805588391499442452548483950930936041319087116164178792915963224051827119629507190496136514154614390306886365249938379780741655336164906300816509165462003900931961194533038028733259054856560208328341977862896624150857506013780599079987035770067431443981872448688326024831024096728625162934978038733692270925546109894719865320919545954364302379422749511450771367972752345949833748922745055355063528194440453870106760351816427592017984114307956025020455600384041819162301558008263800452770020181578243442104131277272722940892151420327446391959611292395593409338396716804354954696359424816542600723856586234593081636718952578459772899512873549731657187799450407370410098681275055795249900366028592041092124481442625111545

Steps: 0

Factorization p: 17503762332113782658072116958792969590172107004036375735420444736728421220935737658128087004838074801181476533575001826642794290112620108806684664313882807227172485515368818233672315897609546729290853787048415449983071727097955613475151491575093295492157765919755561793829693214857805721056057829950873858879472026454384710689192690656674354278485467349305223941260153694737104606520693221358365673196570008301216264172120082403181589809059764520567384671003874108498191393934858866889078386413566549664890319575518501345745782153763330069795593858074251540603595930632699484101018252108339980564312900420337638193003

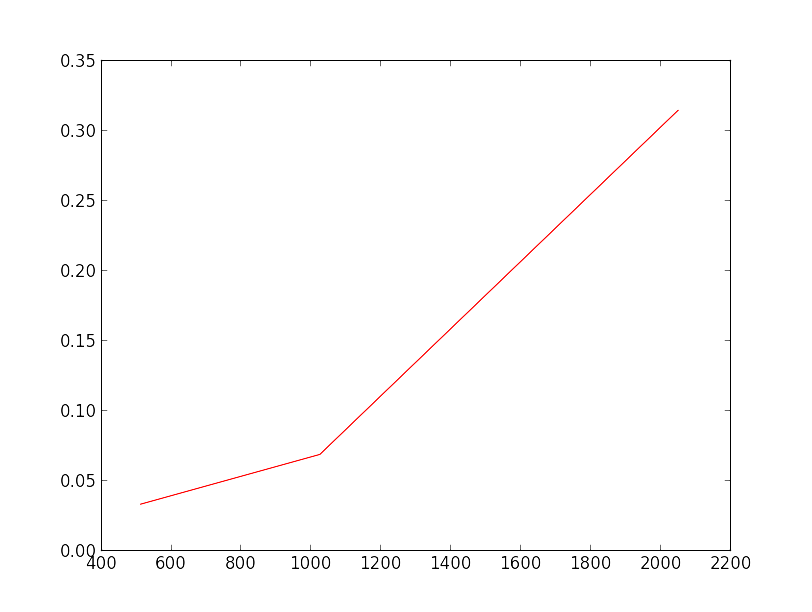
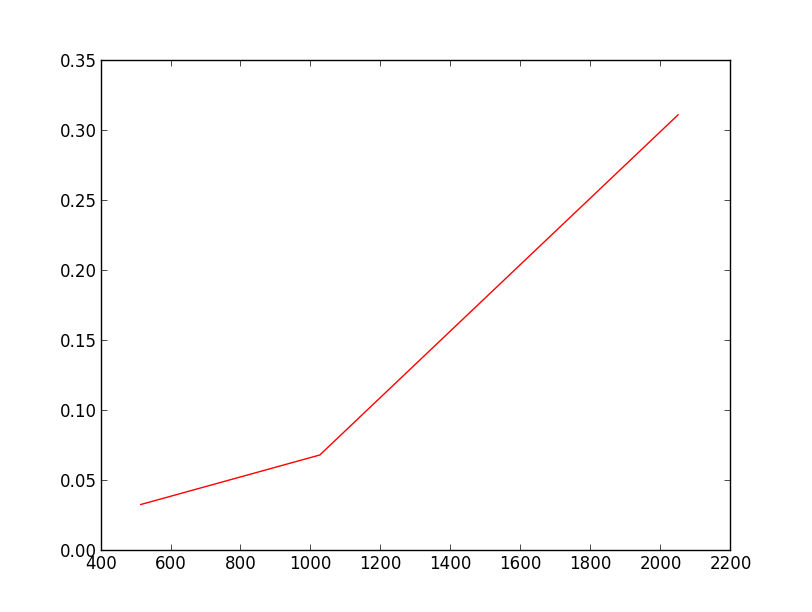
Factorization q: 17503762332113782658072116958792969590172107004036375735420444736728421220935737658128087004838074801181476533575001826642794290112620108806684664313882807227172485515368818233672315897609546729290853787048415449983071727097955613475151491575093295492157765919755561793829693214857805721056057829950873858879472026454384710689192690656674354278485467349305223941260153694737104606520693221358365673196570008301216264172120082403181589809059764520567384671003874108498191393934858866889078386413566549664890319575518501345745782153763330069794593858074251540603595930632699484101018252108339980564312900420337638192073

{'public\_key': {'e': 262145, 'n': 306381695779125327432808059618915510880273889907762024009549947528695792080713367102958856204751601919073984816347958715634835227308556573067575834323990535346915214683971022425626571015468953096289862734958120198053039687953338692474806821888509122865983638547943161556706400172484995711263391082685831264381190047637059582381247891291920624001831887930228974238504619890475241949508460551959634598621522290560411814119868990013728793372272732859285759337443179542122880654691888467415072842370769296314476674638283069630265660246396846732344334584162434155994527154422311698982399031743301968191595349387737251434697273796890206780693046718883975646575118077838974768491557455038215110742060681141644794711568307082453681626994370617813586838002880918138080937982663293752620253352303147201502026748484275664806085252962118968372620880072172769959337138632496618469772890325508831976320450394554348960239250187755180549962162426583533314440044730925579357480014691292943249539323267297815707159720047875747083620681897354797912953604087320288044313662432085892004515863926684921271448430789653321766094770917566238651424712388059207056344946245396899682002113393438044058276872775814429833048618019509490327484361680980327558665219L}, 'private\_key': {'d': 294666156217089231828518571016729663043909032777412764902293548688253873982649048531786186214492661036597501863024141979990349967934771182965039527458463132091779781576369788265926859985851524627933763690325492465823610670302632911119574169796688125144842209213778549409061299349165879966123471449609317225989624124054638078046663486240097349344698000789094810940456706293869837210520981070860077570196100713034319124864923952893441862762252851212939346277501046631976817385575058125438805165428906619451456917684787296336959348936587840298328734020079029017694391198382985171790960828099536651557089439367471069709353545689675126764805889989937672430975520346842056294925633070349434165679726879253477290964854384543050428485138078710117543530294899963977916098579664528418853926087453196253243420904507699948560896926407886067232674646052180802551949113331764811229132711586519781939985947120407054652225795969013085655220994562510271595806132548336169690198656144333555018354402407274794357335881175681996778293486260614786407218512418412280214878011734445233737555028143677128377553739184867460295282824454245144934720194505297736697900778224104150999150993204844327989012893710950108568448701971111936685330540295777868252534865L, 'n': 306381695779125327432808059618915510880273889907762024009549947528695792080713367102958856204751601919073984816347958715634835227308556573067575834323990535346915214683971022425626571015468953096289862734958120198053039687953338692474806821888509122865983638547943161556706400172484995711263391082685831264381190047637059582381247891291920624001831887930228974238504619890475241949508460551959634598621522290560411814119868990013728793372272732859285759337443179542122880654691888467415072842370769296314476674638283069630265660246396846732344334584162434155994527154422311698982399031743301968191595349387737251434697273796890206780693046718883975646575118077838974768491557455038215110742060681141644794711568307082453681626994370617813586838002880918138080937982663293752620253352303147201502026748484275664806085252962118968372620880072172769959337138632496618469772890325508831976320450394554348960239250187755180549962162426583533314440044730925579357480014691292943249539323267297815707159720047875747083620681897354797912953604087320288044313662432085892004515863926684921271448430789653321766094770917566238651424712388059207056344946245396899682002113393438044058276872775814429833048618019509490327484361680980327558665219L}}

Decrypt message: 17503762332113782658072116958792969590172107004036375735420444736728421220935737658128087004838074801181476533575001826642794290112620108806684664313882807227172485515368818233672315897609546729290853787048415449983071727097955613475151491575093295492157765919755561793829693214857805721056057829950873858879472026454384710689192690656674354278485467349305223941260153694737104606520693221358365673196570008301216264172120082403181589809059764520567384671003874108498191393934858866889078386413566549664890319575518501345745782153763330069795593858074251540603595930632699484101018252108339980564312900420337638192903

End test

Test time: 0.311372995377

Test #1

Key's length is 1024

Difference: 10

p: 8225804758707308444460013538213240392289741508755163545267080295413974150177415371526907221295070367815991495644485781543511525999947226175133422552563807

q: 8225804758707308444460013538213240392289741508755163545267080295413974150177415371526907221295070367815991495644485781543511525999947226175133412552563371

Message: 8225804758707308444460013538213240392289741508755163545267080295413974150177415371526907221295070367815991495644485781543511525999947226175133422552563707

Start factorization

Start test

{'public\_key': {'e': 1048577, 'n': 67663863928371800900125801442362219331066180094092923601540744447905116911130477276509539526721162649532103892585708677668440852446379562517455167839471276970787482771043780580368608481068658959995351217896774902136387917114296909625565738011563355805850558305033257910651987569503961948791208031565888513397L}, 'private\_key': {'d': 59991209703826284575977209547728107956410402840207098524249904007627660690969977848111303375435944986590168090983894091622654082149511118194863260553264513834622326202783553810857966380007364375437191393273785630237835808786953983425992348612413058668763208737598066211110365395696003081192394638091519311013L, 'n': 67663863928371800900125801442362219331066180094092923601540744447905116911130477276509539526721162649532103892585708677668440852446379562517455167839471276970787482771043780580368608481068658959995351217896774902136387917114296909625565738011563355805850558305033257910651987569503961948791208031565888513397L}}

Crypt message: 60949761521562757422319519647605619472470578354401001646059620865706681232084203839079081633839178969943646255588423392815801944006604438686250702938958715953981311460834625375562380509397032367078117392885526890834181055876369585922422190471614933549037133948480974762250156842309043672534315056379234905788

Steps: 0

Factorization p: 8225804758707308444460013538213240392289741508755163545267080295413974150177415371526907221295070367815991495644485781543511525999947226175133422552563807

Factorization q: 8225804758707308444460013538213240392289741508755163545267080295413974150177415371526907221295070367815991495644485781543511525999947226175133412552563371

{'public\_key': {'e': 1048577, 'n': 67663863928371800900125801442362219331066180094092923601540744447905116911130477276509539526721162649532103892585708677668440852446379562517455167839471276970787482771043780580368608481068658959995351217896774902136387917114296909625565738011563355805850558305033257910651987569503961948791208031565888513397L}, 'private\_key': {'d': 59991209703826284575977209547728107956410402840207098524249904007627660690969977848111303375435944986590168090983894091622654082149511118194863260553264513834622326202783553810857966380007364375437191393273785630237835808786953983425992348612413058668763208737598066211110365395696003081192394638091519311013L, 'n': 67663863928371800900125801442362219331066180094092923601540744447905116911130477276509539526721162649532103892585708677668440852446379562517455167839471276970787482771043780580368608481068658959995351217896774902136387917114296909625565738011563355805850558305033257910651987569503961948791208031565888513397L}}

Decrypt message: 8225804758707308444460013538213240392289741508755163545267080295413974150177415371526907221295070367815991495644485781543511525999947226175133422552563707

End test

Test time: 0.0332980155945

Test #2

Key's length is 2048

Difference: 10

p: 150028031156876858179788118382881795432379142337955266968913017070376205822334167547049648112018099731155676248518954248603044861917261839091215616553627749002181012746014952541802038552806759189688046704127386685577678963246262829438342792004730896939770501373038903995657389580173360534488286221899664880699

q: 150028031156876858179788118382881795432379142337955266968913017070376205822334167547049648112018099731155676248518954248603044861917261839091215616553627749002181012746014952541802038552806759189688046704127386685577678963246262829438342792004730896939770501373038903995657389580173360534488286221889664880261

Message: 150028031156876858179788118382881795432379142337955266968913017070376205822334167547049648112018099731155676248518954248603044861917261839091215616553627749002181012746014952541802038552806759189688046704127386685577678963246262829438342792004730896939770501373038903995657389580173360534488286221899664880599

Start factorization

Start test

{'public\_key': {'e': 1048577, 'n': 22508410132808813308970059428718674016530114358182232149372462976256801061995242968952752382742952990613378417628433221846692679264685380514596219574645331162303931765055575948780932487180223238550011131220077692436222029274694531496231857309810078206380225134664853197351496573345116843868505209255718530846259624377401325882576564632877498764650039361346090582829099803783526597762199716473412010227629644828925675555607290301028650797794373140441896834599688985833771230802276258093052355521941772277009240860800554136115074851806343431940560914615276052421473527734044411822421422154283861542391980522102684982439L}, 'private\_key': {'d': 6858067603362545396363685534663958070357452277034153285264704639033933961260699191524051031791223773715300135944339929302092115408095287442513374022035040681842614476206903222058103620744313028498949582446975874882291501847715671684321815271469068924987309590305789606315825771705875086376631119417181106795185878081425902371866506344001192349891957816340960963115034047726747701621044616691837241325155968383990503390768948577456181126917176616211567357495746688156060973113629811947991068344272965969837994042987640866237748835296322196954708647410015284963872011696237329502745398163542569606096155768189016028113L, 'n': 22508410132808813308970059428718674016530114358182232149372462976256801061995242968952752382742952990613378417628433221846692679264685380514596219574645331162303931765055575948780932487180223238550011131220077692436222029274694531496231857309810078206380225134664853197351496573345116843868505209255718530846259624377401325882576564632877498764650039361346090582829099803783526597762199716473412010227629644828925675555607290301028650797794373140441896834599688985833771230802276258093052355521941772277009240860800554136115074851806343431940560914615276052421473527734044411822421422154283861542391980522102684982439L}}

Crypt message: 11267036639895064121259832196116766589415380636220319856293888139102268926643445403641136129789765703070050070393698949631363610894685281659647187029279394444276910290043359463910024695003341279145341287403378467604017623764474958883243013224136200512162477746714179743080310455495593898341810149874031567976838441901115919222308418819750960186095898590994471223368826114644235419613707474588826983686972590123883185115870592495314999708539546245887540778548157181949281761971275177647868127288261701276001795347757413271144523203858266217916510557166041948339124152852059111765250534559955478805547867063547755659062

Steps: 0

Factorization p: 150028031156876858179788118382881795432379142337955266968913017070376205822334167547049648112018099731155676248518954248603044861917261839091215616553627749002181012746014952541802038552806759189688046704127386685577678963246262829438342792004730896939770501373038903995657389580173360534488286221899664880699

Factorization q: 150028031156876858179788118382881795432379142337955266968913017070376205822334167547049648112018099731155676248518954248603044861917261839091215616553627749002181012746014952541802038552806759189688046704127386685577678963246262829438342792004730896939770501373038903995657389580173360534488286221889664880261

{'public\_key': {'e': 1048577, 'n': 22508410132808813308970059428718674016530114358182232149372462976256801061995242968952752382742952990613378417628433221846692679264685380514596219574645331162303931765055575948780932487180223238550011131220077692436222029274694531496231857309810078206380225134664853197351496573345116843868505209255718530846259624377401325882576564632877498764650039361346090582829099803783526597762199716473412010227629644828925675555607290301028650797794373140441896834599688985833771230802276258093052355521941772277009240860800554136115074851806343431940560914615276052421473527734044411822421422154283861542391980522102684982439L}, 'private\_key': {'d': 6858067603362545396363685534663958070357452277034153285264704639033933961260699191524051031791223773715300135944339929302092115408095287442513374022035040681842614476206903222058103620744313028498949582446975874882291501847715671684321815271469068924987309590305789606315825771705875086376631119417181106795185878081425902371866506344001192349891957816340960963115034047726747701621044616691837241325155968383990503390768948577456181126917176616211567357495746688156060973113629811947991068344272965969837994042987640866237748835296322196954708647410015284963872011696237329502745398163542569606096155768189016028113L, 'n': 22508410132808813308970059428718674016530114358182232149372462976256801061995242968952752382742952990613378417628433221846692679264685380514596219574645331162303931765055575948780932487180223238550011131220077692436222029274694531496231857309810078206380225134664853197351496573345116843868505209255718530846259624377401325882576564632877498764650039361346090582829099803783526597762199716473412010227629644828925675555607290301028650797794373140441896834599688985833771230802276258093052355521941772277009240860800554136115074851806343431940560914615276052421473527734044411822421422154283861542391980522102684982439L}}

Decrypt message: 150028031156876858179788118382881795432379142337955266968913017070376205822334167547049648112018099731155676248518954248603044861917261839091215616553627749002181012746014952541802038552806759189688046704127386685577678963246262829438342792004730896939770501373038903995657389580173360534488286221899664880599

End test

Test time: 0.0688061714172

Test #3

Key's length is 4096

Difference: 10

p: 26439377549310142888099452455413972166211087472796650133998476349619152092074685267691009974729041919521466156077332214580433178033601594759731642426353270656407566723460728237521735283047031311176067493299321987298967070527240792197381010545991139224836057586066367985209866803805614462012208465006822320130933669509164504348871144558879252528593777045441980675179325682519421168301173181186771277143004836602983148089938212918799416416854380662173599838779368122685238543158305212750934278347630756547520520791042094296453396330431217382878262304654098970214449603315846816686430640362375654495048177475551618364237

q: 26439377549310142888099452455413972166211087472796650133998476349619152092074685267691009974729041919521466156077332214580433178033601594759731642426353270656407566723460728237521735283047031311176067493299321987298967070527240792197381010545991139224836057586066367985209866803805614462012208465006822320130933669509164504348871144558879252528593777045441980675179325682519421168301173181186771277143004836602983148089938212918799416416854380662173599838779368122685238543158305212750934278347630756547520520791042094296453396330431217382878262304654098970214449603315846816686430640362375654495048177475541618364163

Message: 26439377549310142888099452455413972166211087472796650133998476349619152092074685267691009974729041919521466156077332214580433178033601594759731642426353270656407566723460728237521735283047031311176067493299321987298967070527240792197381010545991139224836057586066367985209866803805614462012208465006822320130933669509164504348871144558879252528593777045441980675179325682519421168301173181186771277143004836602983148089938212918799416416854380662173599838779368122685238543158305212750934278347630756547520520791042094296453396330431217382878262304654098970214449603315846816686430640362375654495048177475551618364137

Start factorization

Start test

{'public\_key': {'e': 1048577, 'n': 699040685194965217226293553668768867871956096963518313084995957764400163308479995522216939472994367425663142692035825645036810851011558787919022093514616932479055423916358438284216171212156494042414499735054352230150030886295078794812985278460176062916089237388080032914935742481445761305484169075992779631015170566996017352386373015770044881313033742048722915069445471220988510414193287823775808519972822868879596206265752800028932566342963334207227443527466929260787181495236324975180936717997226039551550346989477069528637416385048457503903847186766152281749618119970522642360547076536833779799106024655561083037418560448244905901641200679334856443402863494831851459584547568766996215247583582554658394277785801369149539243499566275277034912823109857835804450851775378580019441591686595641501318041241249149604986259858152096652661241241156488358393703060036327812158532856100187529505693907218145227141940635177988067753518775976959445784056764371425095099201199663371586619140287104644760116895570857017018358873558335101655101137632302184623824846301859711044557232267250335374237115457943709506327125577369953126466928609880466426417186295803715404884367276853027124742926460026355038365872291072599168450339426080313841638631L}, 'private\_key': {'d': 195070361923729752047380170210693595632951202946941515589032247704690386863048046533307423927086787000309268831107875665777736134889943434705305432775382313948042354154407013148623795733064058931056953154107189082026594649356988572274833056924033349739577419447619105159887492866518953034062088633763898357327453357844683449552847914978559354936267725937987207595122188770088842338042030246815483584924376273428521363634165089274765666458080332891506129021113469178704985115371690444276094071349236452290274483450038371349719280900142792708801837848169131898909432266857440731945407614381626654320108503118477449445833326613335534380458715162753154970129043298309917154388867842578643859589015861285304236019111589732069742378653004718959172365398388339832024342116017738776224018461438432629646874525860111608150337849360099135871217561309873231673781006507554805592681463238756634944981763707641861093134046056100923947710403839649796678923747591835387750172084275004091834962905150917639894449040557268925055055737788082284178940810917030392525282275698271580371793196001112276805613914979105523230573073642273052563231435367363072866240129510298402623570036385454089177639913439466831595309648172464095399914991197635577779412273L, 'n': 699040685194965217226293553668768867871956096963518313084995957764400163308479995522216939472994367425663142692035825645036810851011558787919022093514616932479055423916358438284216171212156494042414499735054352230150030886295078794812985278460176062916089237388080032914935742481445761305484169075992779631015170566996017352386373015770044881313033742048722915069445471220988510414193287823775808519972822868879596206265752800028932566342963334207227443527466929260787181495236324975180936717997226039551550346989477069528637416385048457503903847186766152281749618119970522642360547076536833779799106024655561083037418560448244905901641200679334856443402863494831851459584547568766996215247583582554658394277785801369149539243499566275277034912823109857835804450851775378580019441591686595641501318041241249149604986259858152096652661241241156488358393703060036327812158532856100187529505693907218145227141940635177988067753518775976959445784056764371425095099201199663371586619140287104644760116895570857017018358873558335101655101137632302184623824846301859711044557232267250335374237115457943709506327125577369953126466928609880466426417186295803715404884367276853027124742926460026355038365872291072599168450339426080313841638631L}}

Crypt message: 390728516427412400072129338712637970366418386327528681597817594036291996303896408983098021493894548507484662351453226302153510846760568516316432404851014695113144586216618264551217579167862581216259644364114465836065177251490615093641109881589189033602046937571226454126741218685065603191471562665975195484014124897007466629785127930147389259508507185349127624759673385268997313167862876128100536430201324942565416863656961952223397732664356160224523225334184376498266718935322261257752146891665231813920785938049672527635363242949206478443181461910527579533285537554735218289297069378244896019979658287194651335780703785995216801101536934061660072082019481572213884058527892943400175064576531627445983926099370077063343671209450876464237083607050817821630953418663807486179544944094953891729414965555695193937276441968856603218698746097649410115209936457892016571223898966520589278339041900953427206753979762223857491839641915792630456774417560839889874529842674316518393945436885548290443110168038734065485677600942725196934926839414038307114833259607824339661361997959081414849298880346959045657304084017352241385111415010297637567174157134054433425425668966162921978359313155650312045546731522303018700994173170144438177556950093

Steps: 0

Factorization p: 26439377549310142888099452455413972166211087472796650133998476349619152092074685267691009974729041919521466156077332214580433178033601594759731642426353270656407566723460728237521735283047031311176067493299321987298967070527240792197381010545991139224836057586066367985209866803805614462012208465006822320130933669509164504348871144558879252528593777045441980675179325682519421168301173181186771277143004836602983148089938212918799416416854380662173599838779368122685238543158305212750934278347630756547520520791042094296453396330431217382878262304654098970214449603315846816686430640362375654495048177475551618364237

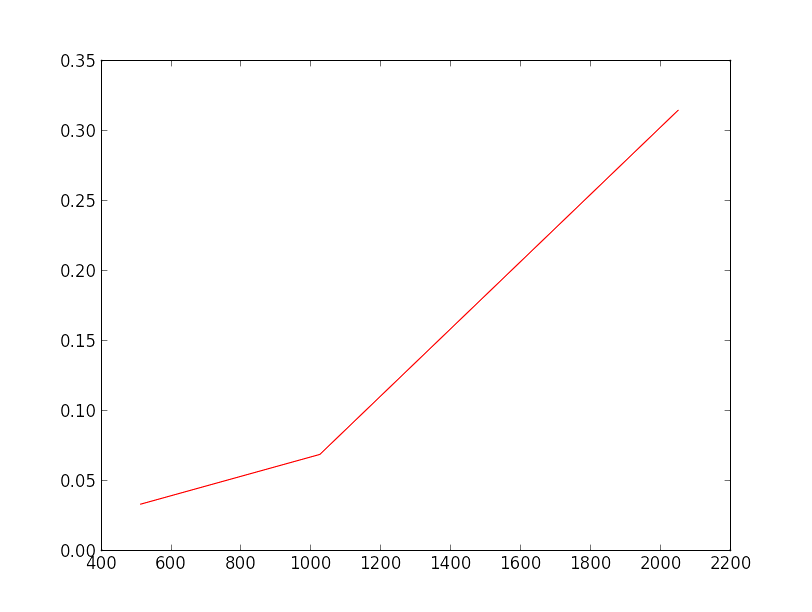
Factorization q: 26439377549310142888099452455413972166211087472796650133998476349619152092074685267691009974729041919521466156077332214580433178033601594759731642426353270656407566723460728237521735283047031311176067493299321987298967070527240792197381010545991139224836057586066367985209866803805614462012208465006822320130933669509164504348871144558879252528593777045441980675179325682519421168301173181186771277143004836602983148089938212918799416416854380662173599838779368122685238543158305212750934278347630756547520520791042094296453396330431217382878262304654098970214449603315846816686430640362375654495048177475541618364163

{'public\_key': {'e': 1048577, 'n': 699040685194965217226293553668768867871956096963518313084995957764400163308479995522216939472994367425663142692035825645036810851011558787919022093514616932479055423916358438284216171212156494042414499735054352230150030886295078794812985278460176062916089237388080032914935742481445761305484169075992779631015170566996017352386373015770044881313033742048722915069445471220988510414193287823775808519972822868879596206265752800028932566342963334207227443527466929260787181495236324975180936717997226039551550346989477069528637416385048457503903847186766152281749618119970522642360547076536833779799106024655561083037418560448244905901641200679334856443402863494831851459584547568766996215247583582554658394277785801369149539243499566275277034912823109857835804450851775378580019441591686595641501318041241249149604986259858152096652661241241156488358393703060036327812158532856100187529505693907218145227141940635177988067753518775976959445784056764371425095099201199663371586619140287104644760116895570857017018358873558335101655101137632302184623824846301859711044557232267250335374237115457943709506327125577369953126466928609880466426417186295803715404884367276853027124742926460026355038365872291072599168450339426080313841638631L}, 'private\_key': {'d': 195070361923729752047380170210693595632951202946941515589032247704690386863048046533307423927086787000309268831107875665777736134889943434705305432775382313948042354154407013148623795733064058931056953154107189082026594649356988572274833056924033349739577419447619105159887492866518953034062088633763898357327453357844683449552847914978559354936267725937987207595122188770088842338042030246815483584924376273428521363634165089274765666458080332891506129021113469178704985115371690444276094071349236452290274483450038371349719280900142792708801837848169131898909432266857440731945407614381626654320108503118477449445833326613335534380458715162753154970129043298309917154388867842578643859589015861285304236019111589732069742378653004718959172365398388339832024342116017738776224018461438432629646874525860111608150337849360099135871217561309873231673781006507554805592681463238756634944981763707641861093134046056100923947710403839649796678923747591835387750172084275004091834962905150917639894449040557268925055055737788082284178940810917030392525282275698271580371793196001112276805613914979105523230573073642273052563231435367363072866240129510298402623570036385454089177639913439466831595309648172464095399914991197635577779412273L, 'n': 699040685194965217226293553668768867871956096963518313084995957764400163308479995522216939472994367425663142692035825645036810851011558787919022093514616932479055423916358438284216171212156494042414499735054352230150030886295078794812985278460176062916089237388080032914935742481445761305484169075992779631015170566996017352386373015770044881313033742048722915069445471220988510414193287823775808519972822868879596206265752800028932566342963334207227443527466929260787181495236324975180936717997226039551550346989477069528637416385048457503903847186766152281749618119970522642360547076536833779799106024655561083037418560448244905901641200679334856443402863494831851459584547568766996215247583582554658394277785801369149539243499566275277034912823109857835804450851775378580019441591686595641501318041241249149604986259858152096652661241241156488358393703060036327812158532856100187529505693907218145227141940635177988067753518775976959445784056764371425095099201199663371586619140287104644760116895570857017018358873558335101655101137632302184623824846301859711044557232267250335374237115457943709506327125577369953126466928609880466426417186295803715404884367276853027124742926460026355038365872291072599168450339426080313841638631L}}

Decrypt message: 26439377549310142888099452455413972166211087472796650133998476349619152092074685267691009974729041919521466156077332214580433178033601594759731642426353270656407566723460728237521735283047031311176067493299321987298967070527240792197381010545991139224836057586066367985209866803805614462012208465006822320130933669509164504348871144558879252528593777045441980675179325682519421168301173181186771277143004836602983148089938212918799416416854380662173599838779368122685238543158305212750934278347630756547520520791042094296453396330431217382878262304654098970214449603315846816686430640362375654495048177475551618364137

End test

Test time: 0.314607143402



**2. Атака на шифрование методом Винера.**

**Test: 1**

Key's length: 32

(41659, 40037)

(40037, 41659)

Test time: 0.425955057144

**Test: 2**

Key's length: 64

(4254529333, 2827332671)

(2827332671, 4254529333)

Test time: 0.343917131424

**Test: 3**

Key's length: 128

(9778497921664574983L, 14985582359886860611L)

(9778497921664574983, 14985582359886860611)

Test time: 1.13723397255

**Test: 4**

Key's length: 256

(219552178111543986144765154893760777067L, 304324160114454464129445121190879974943L)

(219552178111543986144765154893760777067, 304324160114454464129445121190879974943)

Test time: 2.14046692848

**Test: 5**

Key's length: 512

(71028781158690239844361081122692139990083263352158330994297130380148936446821L, 106819780140289609925030215122785840006711610853871753175762967385762316020047L)

(71028781158690239844361081122692139990083263352158330994297130380148936446821, 106819780140289609925030215122785840006711610853871753175762967385762316020047)

Test time: 3.81788396835

**Test: 6**

Key's length: 1024

(10279522491027973392602605959546884722236673928712584804993760953659723997411493914342328579509332799178405945364770157365273826186374679290143756057213831L, 12859521985150434753598278191819279518449120839028988121101993898573017934880803440940111347705463503316374096163856851119720801253265688289728135337191669L)

(10279522491027973392602605959546884722236673928712584804993760953659723997411493914342328579509332799178405945364770157365273826186374679290143756057213831, 12859521985150434753598278191819279518449120839028988121101993898573017934880803440940111347705463503316374096163856851119720801253265688289728135337191669)

Test time: 9.47368717194

**Test: 7**

Key's length: 2048

(115829136379202113702226172332997959016558193530248367537530390067893045286952982905499756082835811956452647653223168018005595179644202322734514369346572730926967849412126072692237882191217272927574486805092584753326502878103706511691503174514200736954366651851071312680271048786120955157749853256912930101821L, 179763404769274633690450129403620606930313207647603690092997431676884017722411836659491688030034452610091127148703224522068755117582410351233753008208541833936088024536075166452314388598141706541355044489935205917582540588723416823963519789823165917782076485200331214526674496200374941875988599496551208612027L)

(115829136379202113702226172332997959016558193530248367537530390067893045286952982905499756082835811956452647653223168018005595179644202322734514369346572730926967849412126072692237882191217272927574486805092584753326502878103706511691503174514200736954366651851071312680271048786120955157749853256912930101821, 179763404769274633690450129403620606930313207647603690092997431676884017722411836659491688030034452610091127148703224522068755117582410351233753008208541833936088024536075166452314388598141706541355044489935205917582540588723416823963519789823165917782076485200331214526674496200374941875988599496551208612027)

Test time: 43.1759319305

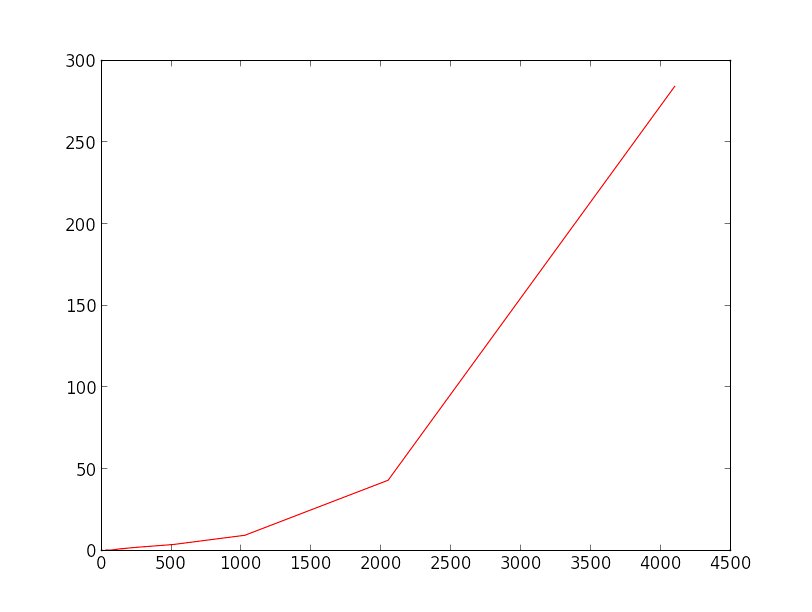
**Test: 8**

Key's length: 4096

(16671909579995247345942609343410797706028164073119489911301140058474603341903257498842040360269876902661082321380356417545994861710272638555573595387592082498262717921021111784663656432534074237349128604938486385120332676515126644817220535272214084621967096653721913158358621400219058046533768104326753263189963357044317954846397970102787152354359675073027744141267092515100856433941363518878742045507802202901969284206075937246234806846119701262858583048534344652964065056328237634507989823047407811029401558227602549553297258421715885608410707033606783488981441244950921928241757794101747379807179542335717303267617L, 29584889151312676636809819117511186416405155060366685790625918200329886021480300174445070209137730340547719855850585200357557431127789243290567275894325505420736511251875860600500782753289394966678648230518342405382617199587534647746939285896175575765054086620911169610527113984366928920440595361970943369603521543640602090990570596906932564387272511870288001279208232799014252420028167953496007952607242444133941149294933903420868831141473663070493100424971415014264868412976823550549125113329882075820387947671411604998142835558076758275246042291948141325024642616767120510215342669565715391317579016479180816809803L)

(16671909579995247345942609343410797706028164073119489911301140058474603341903257498842040360269876902661082321380356417545994861710272638555573595387592082498262717921021111784663656432534074237349128604938486385120332676515126644817220535272214084621967096653721913158358621400219058046533768104326753263189963357044317954846397970102787152354359675073027744141267092515100856433941363518878742045507802202901969284206075937246234806846119701262858583048534344652964065056328237634507989823047407811029401558227602549553297258421715885608410707033606783488981441244950921928241757794101747379807179542335717303267617, 29584889151312676636809819117511186416405155060366685790625918200329886021480300174445070209137730340547719855850585200357557431127789243290567275894325505420736511251875860600500782753289394966678648230518342405382617199587534647746939285896175575765054086620911169610527113984366928920440595361970943369603521543640602090990570596906932564387272511870288001279208232799014252420028167953496007952607242444133941149294933903420868831141473663070493100424971415014264868412976823550549125113329882075820387947671411604998142835558076758275246042291948141325024642616767120510215342669565715391317579016479180816809803)

Test time: 284.323810101

**3. Анализ скорости работы метода атаки повторным шифрованием.**

Проверялось на ключах, которые генерировались в соответствии с плохими ключами для атаки Винера.

**Test 1:**

Key length is 16

Test time: 7.39097595215e-05

**Test 2:**

Key length is 17

Test time: 6.29425048828e-05

**Test 3:**

Key length is 18

Test time: 8.01086425781e-05

**Test 4:**

Key length is 19

Test time: 0.000241041183472

**Test 5:**

Key length is 20

Test time: 0.00582599639893

**Test 6:**

Key length is 21

Test time: 0.000379085540771

**Test 7:**

Key length is 22

Test time: 0.0279068946838

**Test 8:**

Key length is 23

Test time: 0.0106627941132

**Test 9:**

Key length is 24

Test time: 0.0260360240936

**Test 10:**

Key length is 25

Test time: 0.0387771129608

**Test 11:**

Key length is 26

Test time: 0.00180006027222

**Test 12:**

Key length is 27

Test time: 0.00105595588684

**Test 13:**

Key length is 28

Test time: 0.339407920837

**Test 14:**

Key length is 29

Test time: 4.15881299973

**Test 15:**

Key length is 30

Test time: 32.1181590557

**Test 16:**

Key length is 31

Test time: 3.7442419529

**Test 17:**

Key length is 32

Test time: 0.212326049805

**Test 18:**

Key length is 33

Test time: 10.6967539787

**Test 19:**

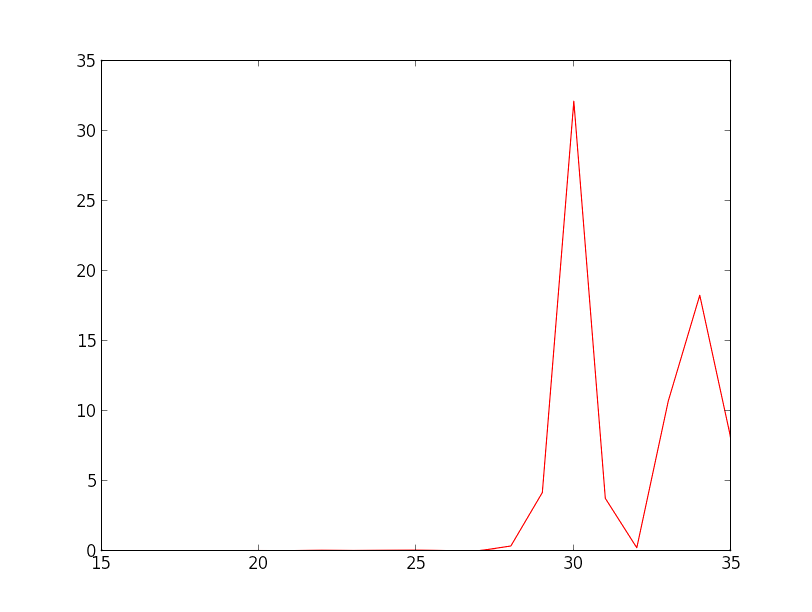
Key length is 34

Test time: 18.2520501614

**Test 20:**

Key length is 35

Test time: 7.88876605034

Можно сделать вывод, что от размера ключа зависимость присутствует, но по большей мере она обусловлена выбором неудачных коэффициентов.

**4. Атака на основе китайской теоремы об остатках.**

При использовании e = 3

**Test0**

Key length is 16

456

Test time: 4.31537628174e-05

**Test1**

Key length is 32

456

Test time: 3.09944152832e-05

**Test2**

Key length is 64

456

Test time: 3.09944152832e-05

**Test3**

Key length is 128

456

Test time: 3.00407409668e-05

**Test4**

Key length is 256

456

Test time: 4.10079956055e-05

**Test5**

Key length is 512

456

Test time: 3.91006469727e-05

**Test6**

Key length is 1024

456

Test time: 4.19616699219e-05

**Test7**

Key length is 2048

456

Test time: 5.10215759277e-05

Из этого можно сделать вывод, что атака практически не зависит от длины ключа.

**Исходный код.**

# -\*- coding: utf-8 -\*-

import math

import sys

import time

from standart\_rsa import generate\_rsa\_key, rsa\_decrypt, rsa\_encrypt, generate\_bad\_rsa\_key

from prime\_num import Zpow, generate\_prime\_fix\_len, miller\_rabin\_test, AlgEvklid\_ex, AlgEvklid

import gmpy2

from WienerAttack import WienerAttack

import pylab

import random

from sympy.solvers import solve

from sympy import Symbol

class BreakingRSA:

@staticmethod

def factorization(n, delta):

gmpy2.get\_context().precision=1000000

t = gmpy2.mpz(gmpy2.sqrt(n))

res = gmpy2.mpfr('0.0', 1000000)

steps = delta-t

print "Steps: %d" % steps

while True:

res = gmpy2.sqrt(t\*\*2 - n)

if gmpy2.is\_integer(res):

break

if res >= n/2:

sys.stderr.write('I can\'t solve')

exit()

t += 1

steps -= 1

if steps % 1000 == 0:

print steps

p = t - gmpy2.mpz(res)

q = n/p

return {

'p': int(q),

'q': int(p)

}

@staticmethod

def repeat\_attack(e, n, mes):

res = Zpow(mes, e, n)

pred = None

while res != mes:

pred = res

res = Zpow(res, e, n)

return pred

@staticmethod

def chines\_theorem\_attack(e, users\_n, cr):

m1 = users\_n[1]\*users\_n[2]

m2 = users\_n[0]\*users\_n[2]

m3 = users\_n[0]\*users\_n[1]

m1\_inv = AlgEvklid\_ex(users\_n[0], m1)['y']

m2\_inv = AlgEvklid\_ex(users\_n[1], m2)['y']

m3\_inv = AlgEvklid\_ex(users\_n[2], m3)['y']

n1 = m1\_inv % users\_n[0]

n2 = m2\_inv % users\_n[1]

n3 = m3\_inv % users\_n[2]

S = cr[0]\*n1\*m1 + cr[1]\*n2\*m2 + cr[2]\*n3\*m3

res = S % (users\_n[0]\*users\_n[1]\*users\_n[2])

res = int((res \*\* (1/3.0))+0.5)

return res

@staticmethod

def nokey\_reading(e1, e2, N, mes1, mes2):

koef = AlgEvklid\_ex(e1, e2)

r = koef['x']

s = koef['y']

x = (gmpy2.powmod(mes1, r, N)\*gmpy2.powmod(mes2, s, N)) % N

return x

@staticmethod

def test\_factorization(p, q, mes):

print "Start test"

bits\_len = len(str(bin(p)))-2

key = generate\_rsa\_key(p, q, bits\_len)

#fkey = open(infile+'key', 'w').write(key)

print key

res = Zpow(mes, key['public\_key']['e'], key['public\_key']['n'])

print "Crypt message: %d" % res

delta = (p\*q+p\*p)/2/p

nums = BreakingRSA.factorization(key['public\_key']['n'], delta)

print "Factorization p: %s" % nums['p']

print "Factorization q: %s" % nums['q']

bits\_len = len(str(bin(nums['p'])))-2

rekey = generate\_rsa\_key(nums['p'], nums['q'], bits\_len)

print rekey

res = Zpow(res, rekey['private\_key']['d'], rekey['private\_key']['n'])

print "Decrypt message: %d" % res

if mes != res:

sys.stderr.write('Fatal')

exit()

print "End test"

@staticmethod

def test\_repeat\_attack():

# n = 85517

# e = 397

# mes = 8646

power = 16

i = 1

mes = 124

xlist = []

ylist = []

while power < 36:

print "Test %d:" % i

res = None

while res is None:

key = WienerAttack.create\_rsa\_key(power)

crypt = Zpow(mes, key['public\_key']['e'], key['public\_key']['n'])

start = time.time()

res = BreakingRSA.repeat\_attack(key['public\_key']['e'], key['public\_key']['n'], crypt)

finish = time.time()

if mes != res:

sys.stderr.write("Error. Can't decrypt message")

exit()

print "Key length is %s" % power

print "Test time: %s" % (finish - start)

xlist.append(power)

ylist.append(finish-start)

power += 1

i += 1

pylab.plot(xlist, ylist, 'r')

pylab.show()

@staticmethod

def test\_chines\_theorem\_attack():

users\_n = []

cr = []

mes = 456

power = 16

i = 0

e = 3

while power < 4000:

print "Test%d" % i

print "Key length is %d" % power

phi = []

for i in xrange(3):

q = generate\_prime\_fix\_len(power)

while q is None:

q = generate\_prime\_fix\_len(power)

p = generate\_prime\_fix\_len(power)

while p is None:

p = generate\_prime\_fix\_len(power)

users\_n.append(p\*q)

phi.append((p-1)\*(q-1))

pos = 8

for i in xrange(3):

cr.append(Zpow(mes, e, users\_n[i]))

start = time.time()

print BreakingRSA.chines\_theorem\_attack(e, users\_n, cr)

finish = time.time()

print "Test time: %s" % (finish-start)

power \*= 2

i += 1

class RSAAttackTest:

@staticmethod

def test\_nokey\_reading():

N = 137759

e1 = 191

e2 = 233

y1 = 60197

y2 = 63656

print BreakingRSA.nokey\_reading(e1, e2, N, y1, y2)

if \_\_name\_\_ == "\_\_main\_\_":

#print BreakingRSA.factorization(23360947609)

#BreakingRSA.test\_repeat\_attack()

BreakingRSA.test\_chines\_theorem\_attack()

#RSAAttackTest.test\_nokey\_reading()

#print BreakingRSA.wienerAttack(90581, 17993)

# power = 512

# i = 1

# pos = 10

# xlist = []

# ylist = []

# while power < 4000:

# print "Test #%d" % i

# print "Key's length is %d" % (power\*2)

#

# p = None

# q = None

# while True:

# q = generate\_prime\_fix\_len(power) # generate 2nd prime

#

# p = q + 10\*\*pos

# while not miller\_rabin\_test(p, 30):

# p += 1

#

# if q < p and q is not None and p is not None:

# # str\_q = str(q)

# # str\_p = str(p)

# # pos = 0

# # for j in xrange(len(str\_q)+1):

# # if str\_q[-j] != str\_p[-j]:

# # pos = j

# # if pos != 3:

# # continue

# break

#

# print "Difference: %d" % pos

# print "p: %s" % p

# print "q: %s" % q

#

# mes = p - 100

# print "Message: %s" % mes

# print "Start factorization"

# start = time.time()

# BreakingRSA.test\_factorization(p, q, mes)

# finish = time.time()

# print "Test time: %s" % (finish - start)

# xlist.append(power)

# ylist.append(finish-start)

# power \*= 2

#

# i += 1

#

# pylab.plot(xlist, ylist, 'r')

# pylab.show()

# -\*- coding: utf-8 -\*-

import math

import sys

import time

from standart\_rsa import generate\_rsa\_key, rsa\_decrypt, rsa\_encrypt, generate\_bad\_rsa\_key

from prime\_num import Zpow, generate\_prime\_fix\_len, miller\_rabin\_test, AlgEvklid\_ex

import gmpy2

import random

from sympy.solvers import solve

from sympy import Symbol

import pylab

class WienerAttack:

@staticmethod

def create\_rsa\_key(size=2048):

p = generate\_prime\_fix\_len(size // 2)

while p is None:

p = generate\_prime\_fix\_len(size // 2)

q = generate\_prime\_fix\_len(size // 2)

while q is None:

q = generate\_prime\_fix\_len(size // 2)

N = p \* q

phiN = (p-1)\*(q-1)

while True:

d = random.getrandbits(size // 5)

try:

e = int(gmpy2.invert(d, phiN))

except:

continue

if (e \* d) % phiN == 1:

break

public\_key = {

'e': e,

'n': N

}

private\_key = {

'd': d,

'N': N

}

return {

'public\_key': public\_key,

'private\_key': private\_key

}

@staticmethod

def makeNextFraction(fraction):

(a, b) = fraction

res = b / a

a1 = b % a

b1 = a

return res, (a1, b1)

@staticmethod

def makeContinuedFraction(fraction):

(a, b) = fraction

v = []

v.append(0)

while not a == 1:

r, fraction = WienerAttack.makeNextFraction(fraction)

(a, b) = fraction

v.append(r)

v.append(b)

return v

@staticmethod

def makeIndexedConvergent(sequence, index):

(a, b) = (1,sequence[index])

while index>0:

index -= 1

(a, b) = (b, sequence[index]\*b+a)

return (b, a)

@staticmethod

def makeConvergents(sequence):

r = []

for i in xrange(0, len(sequence)):

r.append(WienerAttack.makeIndexedConvergent(sequence, i))

return r

@staticmethod

def solveQuadratic(a, b, c):

x = Symbol('x')

return solve(a\*x\*\*2 + b\*x + c, x)

@staticmethod

def wienerAttack(N, e):

conv = WienerAttack.makeConvergents(WienerAttack.makeContinuedFraction((e, N)))

for frac in conv:

(k, d) = frac

if k == 0:

continue

phiN = ((e\*d)-1)/k

roots = WienerAttack.solveQuadratic(1, -(N-phiN+1), N)

if len(roots) == 2:

p, q = roots[0] % N, roots[1] % N

if p\*q == N:

return p, q

if \_\_name\_\_ == "\_\_main\_\_":

size = 32

i = 1

xlist = []

ylist = []

while size < 5000:

print "Test: %d" % i

print "Key's length: %d" % size

key = WienerAttack.create\_rsa\_key(size)

start = time.time()

factorization = WienerAttack.wienerAttack(key['public\_key']['n'], key['public\_key']['e'])

finish = time.time()

print factorization

print "Test time: %s" % (finish-start)

xlist.append(size)

ylist.append(finish-start)

size \*= 2

i += 1

pylab.plot(xlist, ylist, 'r')

pylab.show()

**Выводы.**

1. Все атаки имеют право на существование. Наиболее плохой атакой является атака повторным шифрованием. При больших ключах ее использование мало вероятно. Атака Винера одна из самых действенных и быстрых атак.
2. При реализации шифра RSA и выборе коэффициентов необходимо учитывать множество нюансов, так как все это может сказаться на конечность защищенности системы.