# Questions on RSA video

## Big picture questions

Why not simply reverse the steps?

| |
|---|
| if any one can see the cipher text, and the elements of the public key, wouldn't they just be able to decrypt the cipher text to view the original message by doing the steps in reverse? or is the 'd' value (stated in the video) really the only thing that will garrontee you to get the original message back? |
| Is there a way to reverse the client key to recover the data with the knowledge of the power and modulus? Would someone be able to guess the multiple of the divisor that was taken off by the modular function? |
| how difficult is it to reverse engineer an RSA public key? at what point is it computationally unfeasible to factor p and q? |

Why not factor n to get p and q?

| |
|---|
| If you know n, because Bob makes it public, can't you find z, which Bob wants to remain private? Because n is the product of two prime numbers, I would think an algorithm to find z, the product of one less than the primes (which are less than n) that multiply to get n wouldn't be hard. Unless real-life ns and zs and ps and qs are all so large that it'd take a supercomputer to figure out z. |
| How easily can P and Q (the numbers that have the prime factors) be obtained by a potential hacker? |
| Why is factoring such a large product of two primes completely infeasible? It would take a long time, but couldn't a really powerful computers (or several computers working together) for a long time eventually find the prime combinations and get the private key? |
| At first I was wondering how Trudy couldn't just reverse engineer the values until I realized how much work it would take to factor such a large number into its two prime factors |
| Why is it difficult for intruders to find the prime numbers of the key? |

What resources do you need to factor an 8-bit, 16-bit, 1024-bit, 2048-bit etc. key?

| |
|---|
| [Repeated] If you know n, because Bob makes it public, can't you find z, which Bob wants to remain private? Because n is the product of two prime numbers, I would think an algorithm to find z, the product of one less than the primes (which are less than n) that multiply to get n wouldn't be hard. Unless real-life ns and zs and ps and qs are all so large that it'd take a supercomputer to figure out z. |
| [Repeated] How easily can P and Q (the numbers that have the prime factors) be obtained by a potential hacker? |
| [Repeated] Why is factoring such a large product of two primes completely infeasible? It would take a long time, but couldn't a really powerful computers (or several computers working together) for a long time eventually find the prime combinations and get the private key? |
| [Repeated] At first I was wondering how Trudy couldn't just reverse engineer the values until I realized how much work it would take to factor such a large number into its two prime factors |
| Why do primes p and q need to be large numbers?<br>Does the size of p and q change how long prime factorization takes?<br>And why do p and q need to be close to each other? |
| For 8-bit, 16-bit encryption, and on, what would be the computer specifications that would be unable to handle each level of encryption, and what part of the computer really determines that? |
| [Repeated] Why is it difficult for intruders to find the prime numbers of the key? |
| Is there any way to tell how good your encryption is? (aka, how easy it would be for someone else to figure out how to solve it?) |

Has anyone ever done it before?

| |
|---|
| Has anyone learned how to reverse engineer a key using RSA before? |

Is the d value really the only way to recover the message?

| |
|---|
| [Repeated] if any one can see the cipher text, and the elements of the public key, wouldn't they just be able to decrypt the cipher text to view the original message by doing the steps in reverse? or is the 'd' value (stated in the video) really the only thing that will garrontee you to get the original message back? |

Why do p and q need to be close to each other?

| |
|---|
| [Repeated] Why do primes p and q need to be large numbers?<br>Does the size of p and q change how long prime factorization takes?<br>And why do p and q need to be close to each other? |

Future of RSA

| |
|---|
| Can RSA encryption become obsolete by brute force in the future? Like when we have more powerful machines, does brute force ever become an option for this? or will everything remain secure in the future? |
| If it someday became possible to factor find p and q from n, how else would we encrypt data? (since this way would become much less secure) |

How can we find p and q efficiently? [Beyond scope]

| |
|---|
| Since RSA requires very large primes to make it properly secure, there is very large math being done on the numbers. In the video you showed that it is possible to make these numbers smaller using modulus, because that is the only part of the number that matters. After lab today we were talking about how the first many digits of the number are ignored when finding the modulus to make it faster. So...<br><br>1. How do you ignore the first many digits of a number in math while still ensuring that there are enough digits to accurately calculate the modulus?<br>2. How many digits of precision are actually required to calculate the modulus of a number? |

Other big-picture questions

| |
|---|
| How would an intruder acquire a private key? |
| [Repeated] Is there any way to tell how good your encryption is? (aka, how easy it would be for someone else to figure out how to solve it?) |
| Would you be more secure if you used two private keys instead of public and private? And just share 1? |
| What is the most common application of RSA encryption in the real world? |

How is this applied? How do people come up with this stuff?

## All questions

Recovering the key

| |
|---|
| if any one can see the cipher text, and the elements of the public key, wouldn't they just be able to decrypt the cipher text to view the original message by doing the steps in reverse? or is the 'd' value (stated in the video) really the only thing that will garrontee you to get the original message back? |
| If you know n, because Bob makes it public, can't you find z, which Bob wants to remain private? Because n is the product of two prime numbers, I would think an algorithm to find z, the product of one less than the primes (which are less than n) that multiply to get n wouldn't be hard. Unless real-life ns and zs and ps and qs are all so large that it'd take a supercomputer to figure out z. |
| How easily can P and Q (the numbers that have the prime factors) be obtained by a potential hacker? |
| Why is factoring such a large product of two primes completely infeasible? It would take a long time, but couldn't a really powerful computers (or several computers working together) for a long time eventually find the prime combinations and get the private key? |
| At first I was wondering how Trudy couldn't just reverse engineer the values until I realized how much work it would take to factor such a large number into its two prime factors |
| Why do primes p and q need to be large numbers?<br>Does the size of p and q change how long prime factorization takes?<br>And why do p and q need to be close to each other? |
| For 8-bit, 16-bit encryption, and on, what would be the computer specifications that would be unable to handle each level of encryption, and what part of the computer really determines that? |
| Why is it difficult for intruders to find the prime numbers of the key? |
| Is there a way to reverse the client key to recover the data with the knowledge of the power and modulus? Would someone be able to guess the multiple of the divisor that was taken off by the modular function? |
| how difficult is it to reverse engineer an RSA public key? at what point is it computationally unfeasible to factor p and q? |
| Has anyone learned how to reverse engineer a key using RSA before? |

Finding keys efficiently

| |
|---|
| Since RSA requires very large primes to make it properly secure, there is very large math being done on the numbers. In the video you showed that it is possible to make these numbers smaller using modulus, because that is the only part of the number that matters. After lab today we were talking about how the first many digits of the number are ignored when finding the modulus to make it faster. So...<br><br>1. How do you ignore the first many digits of a number in math while still ensuring that there are enough digits to accurately calculate the modulus?<br>2. How many digits of precision are actually required to calculate the modulus of a number? |

Future of RSA

| |
|---|
| Can RSA encryption become obsolete by brute force in the future? Like when we have more powerful machines, does brute force ever become an option for this? or will everything remain secure in the future? |
| If it someday became possible to factor find p and q from n, how else would we encrypt data? (since this way would become much less secure) |

Other big-picture questions

| |
|---|
| How would an intruder acquire a private key? |
| Is there any way to tell how good your encryption is? (aka, how easy it would be for someone else to figure out how to solve it?) |
| Would you be more secure if you used two private keys instead of public and private? And just share 1? |
| What is the most common application of RSA encryption in the real world? |

How is this applied? How do people come up with this stuff?

Reusing private key

| |
|---|
| When a new message is sent does Bob need to choose a new private key? Or does he stay with the same private me for all of his messages? |

How to use with non-numeric messages?

| |
|---|
| How does RSA encryption work for messages that aren't simply numbers? |
| How would you encode a string with this? would you have to turn all the letters into ascii numbers and then encode each one? |
| How would you encrypt text data using this, or in other words how would you convert your text into numbers. |
| In the example the message 'm' is a number, but how does the encryption work with real messages like files? |
| So then RSA can be used to encrypt values less than the product of the 2 primes? Then it cuts messages into blocks so that the maximum value of each block is less than the product of the primes, and these blocks are just appended to one another? |

Memorizing the steps

| |
|---|
| The steps dont seem so crazy when laid out, but I can't seem to remember them without looking. Any tips for that? |
| Will we have to be able to do any of these computations are proofs? |

Algorithm

| |
|---|
| Where does p and q come from? Are they random numbers? |
| [Repeated] Why do primes p and q need to be large numbers?<br>Does the size of p and q change how long prime factorization takes?<br>And why do p and q need to be close to each other? |
| What is the general rule for size of primes in rsa besides them being close to each other? By this i mean in number of digits in length. |
| i watched the video multiple times and i am still a little confused on the calculations. |
| How did they pick which letters to use with what? It seems a little confusing |
| I'm having a hard time wrapping my head around the variables used in the video. Can we do an example in class of encrypting and decrypting? Also the video used the number 65537 as e. Can this be any number? What are the pros and cons of using different numbers. Does it change anything? |

Numeric examples in the video

| |
|---|
| Early on the example for n and z were 7 and 3 respectively. But, using the equations n = p*q and z = (p-1)(q-1) doesn't work out well for those numbers as an example as far as I can tell. If Trudy has the public key, then wouldn't they know what e and n were? |
| d=27 isn't the "only" number that satisfies d*e=1 mod40, but it is the first. Is there any benefit to using the next smallest number? Could this have an opposite effect, making the private key less secure by exposing a private value? |
| [Repeated] i watched the video multiple times and i am still a little confused on the calculations. |
| i am confused the power part when the private and public key translate. |

Mathematical theory

| |
|---|
| how does the mod z actually work with the public and private key |
| why do the numbers we use for p and q have to be prime? |
| Wouldn't it be that if your message, raised to the exponent, is shorter than the primes, that your message would be just plain text as it would be the remainder? |
| Why must p and q be prime? Why won't it work with numbers that are not prime? |
| Why are prime numbers more effective than other types of numbers when creating the p and q values? |

Other

| |
|---|
| So from I got from the video, we need two large primes.<br><br>p = some large prime<br>q = some other large prime<br>and n = p*q<br><br>We also need a random number and d to be its modular inverse<br><br>e = some random number<br>d = the modular inverse based off of a another number z to e<br><br>and with that, the key that we make is Ke = (e,n)<br>Kd = (d,n) and no one knows this.<br><br>Could you go through another analogy for us? |
| What are modulus in key and how to know private exponent is e and not d? |