RSA Talk

Eagle

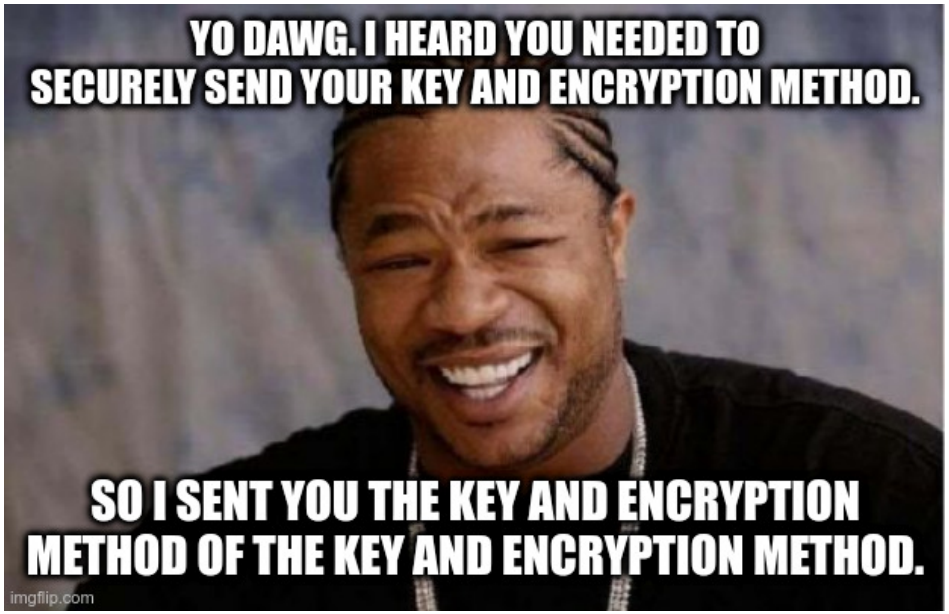
August 17, 2023

Agenda

1. Introduction
   - Cryptography
   - Brief History of RSA
   - Purpose of RSA
   - Relativity of RSA
2. RSA formulas
3. Modulus
4. Totients
5. Variables
6. Example
7. Vulnerabilities
   - Small Public Key
   - Small Private Key
   - Modulus

Cryptography Introduction

- There was one caveat to all the cryptosystems before RSA: they were all based on the fact that both the decoding and encoding parties had to know the method of encryption and the key to decrypting the cipher.



- Up until the 20th century, key distribution remained a problem. Correct key and encryption methods had to be relayed securely, but how does one relay the key securely? Why, with encryption, of course. But, then how do you send the key and encryption method to the key and encryption method of the message?
- As you can see, this caveat of cryptography is an infinite loop between encryption and key distribution that cannot be solved with security ensured for all the desired readers.

History of RSA

1. Whitfield Diffie and his partner Martin Hellman attempted to create an asymmetrical cipher using a one-way function, however they failed.
2. The first creation of an encryption scheme like RSA was created by Clifford Cocks, a mathematician working for the British GCHQ in 1973. It was never deployed because of the price of computers at the time, and it was later declassified in 1977.
3. In April of 1977, MIT students Ron Rivest, Adi Shamir, and Leonard Adleman, developed the famous encryption.
    1. They spent Passover at a student's house and drank too much Manischewitz.
    2. Rivest, as all people do in their drunken states, started thinking about their one-way function, and had much of the paper ready by daybreak.
    3. Rivest had the main breakthrough, but do not forget the importance of Adleman's flaw spotting and Shamir's collaboration with Rivest to develop a mathematical one-way function.

Fun Fact

RSA is an acronym of the creators' last names (Rivest–Shamir–Adleman).

Purpose of RSA

RSA serves as a public key cryptography system that secures data over the internet. It is used in emails, ssl certificates, digital signatures, VPNs, communication, TLS handshakes, and so much more.

Important

RSA itself is secure when used correctly. We will be exploiting improper implementation by people who don't know how RSA works.

Relativity of RSA

- RSA is based on the fundamental principle that factorization is harder than multiplying two numbers together.

Relativity of RSA

- RSA is based on the fundamental principle that factorization is harder than multiplying two numbers together.
- For example, can you multiply 17 by 13?

Relativity of RSA

- RSA is based on the fundamental principle that factorization is harder than multiplying two numbers together.
- For example, can you multiply 17 by 13?
- Easy enough. The answer is 221.
- Now try to find the two factors of the product 187.

Relativity of RSA

- RSA is based on the fundamental principle that factorization is harder than multiplying two numbers together.
- For example, can you multiply 17 by 13?
- Easy enough. The answer is 221.
- Now try to find the two factors of the product 187.
- The answers are 17 and 11. Notice how factorization is significantly harder than multiplication?

Relativity of RSA

- RSA is based on the fundamental principle that factorization is harder than multiplying two numbers together.
- For example, can you multiply 17 by 13?
- Easy enough. The answer is 221.
- Now try to find the two factors of the product 187.
- The answers are 17 and 11. Notice how factorization is significantly harder than multiplication?
- That is precisely how RSA operates. By producing a large product from two primes, the factorization of that number will take significantly longer than the time it took to create it. This is called a "trapdoor function".

Relativity of RSA

- RSA is based on the fundamental principle that factorization is harder than multiplying two numbers together.
- For example, can you multiply 17 by 13?
- Easy enough. The answer is 221.
- Now try to find the two factors of the product 187.
- The answers are 17 and 11. Notice how factorization is significantly harder than multiplication?
- That is precisely how RSA operates. By producing a large product from two primes, the factorization of that number will take significantly longer than the time it took to create it. This is called a "trapdoor function".
- Thus, it is important to note that RSA is only as secure as the current technology. An impossible factorization 10 years ago may be cracked instantly today. It is relative, and thus a generated algorithm gets less and less secure by the second as processing power becomes faster.

Formulas

Main Formulas

$$C \equiv M^e \ mod \ n \tag{1}$$

$$M \equiv C^d \ mod \ n \tag{2}$$

$$de \equiv 1 \ mod \ \phi(n) \tag{3}$$

Additional Formulas

$$n = pq \tag{4}$$

$$\phi(n) = (p-1) \times (q-1) \tag{5}$$

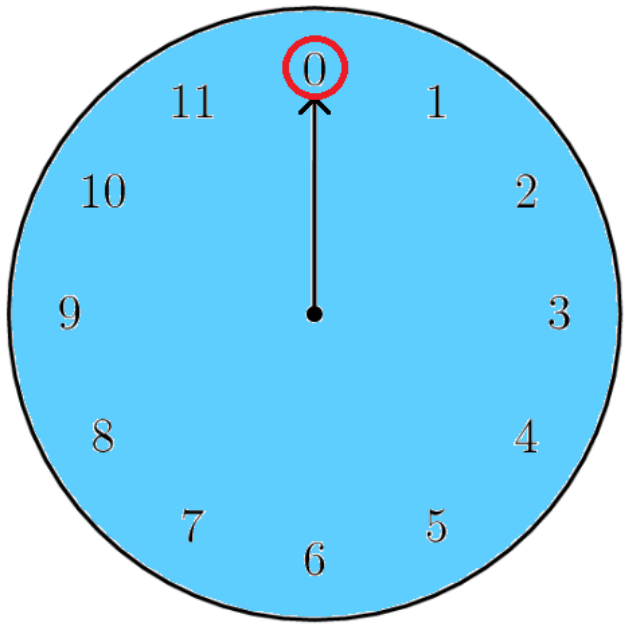
M = plaintext
C = ciphertext
e = public key exponent
d = private key exponent
n = modulus

Modular Arithmetic

Modular arithmetic is a way of doing basic subtraction, multiplication, etc. with integers in a special way.
Although modular arithmetic is a concept in of itself, I plan on going over the modular multiplicative inverse and other necessary terms to understand RSA.

Real World Use

A clock is a classic example of modular arithmetic in real life. Notice how the hour never gets larger than 12, instead looping back to 1. That is how modulus works.

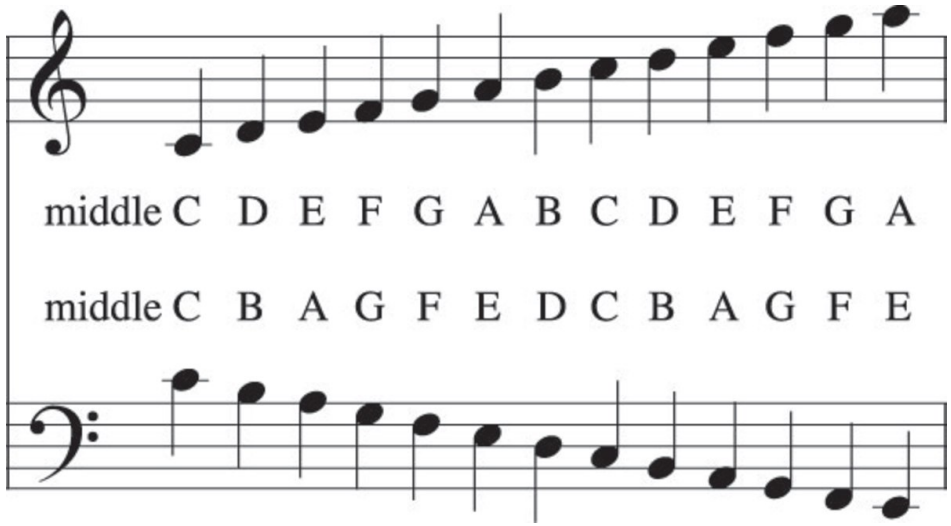


Example 1: If the time is 1:00, what will the time be in 30 hours?

Real World Use

Another example of modern modular arithmetic is the music scale. The treble clef starts at middle C, progresses through G, and loops to A, then G, etc.

0	1	2	3	4	5	6
A	B	C	D	E	F	G

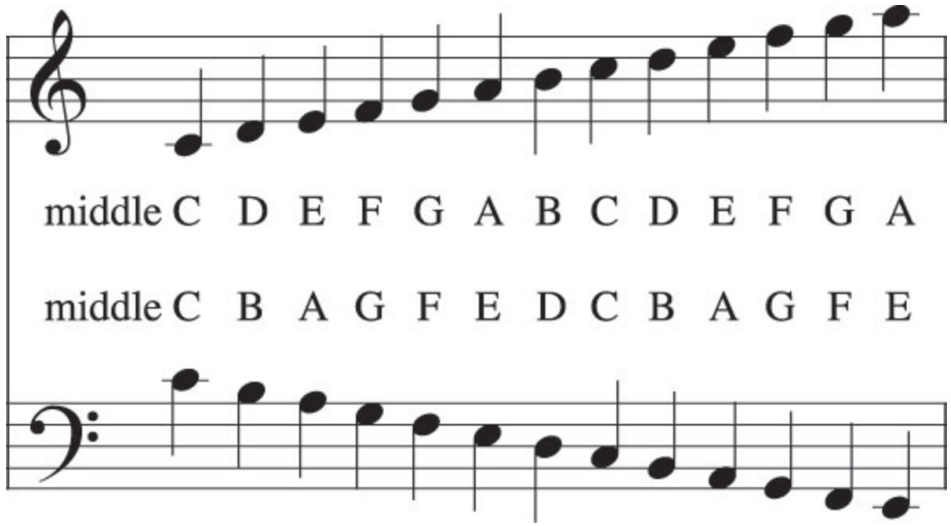


Example 1: If I am playing an F key at the bottom of a piano, what key am I playing 20 white keys away?

Real World Use

Another example of modern modular arithmetic is the music scale. The treble clef starts at middle C, progresses through G, and loops to A, then G, etc.

0	1	2	3	4	5	6
A	B	C	D	E	F	G



Example 1: If I am playing an F key at the bottom of a piano, what key am I playing 20 white keys away?

Answer: The modulus of a scale is 7 notes, and F corresponds to 5, so $x = (5 + 20) \bmod 7$. Split up, $x = (7 \times 3) + 4 \bmod 7$. So, you will be playing E 20 keys away from F.

Congruent Modulo

Congruent modulo equation

$$a \equiv b (\text{mod } n) \tag{6}$$

a and b are congruent modulo n if they share the same remainders when divided by n. (If its modulus on one side, its modulus on the other).

Example 1: $5 \equiv 15$ mod 10 because the remainder of $15 \div 10$ and $5 \div 10$ are 5.

Congruent Modulo

Congruent modulo equation

$$a \equiv b (\text{mod } n) \tag{6}$$

a and b are congruent modulo n if they share the same remainders when divided by n. (If its modulus on one side, its modulus on the other).

Example 1: $5 \equiv 15 \text{ mod } 10$ because the remainder of $15 \div 10$ and $5 \div 10$ are 5.
Example 2: What is a if $a \equiv 12 \text{ mod } 7$?

Congruent Modulo

Congruent modulo equation

$$a \equiv b (\text{mod } n) \tag{6}$$

a and b are congruent modulo n if they share the same remainders when divided by n. (If its modulus on one side, its modulus on the other).

Example 1: $5 \equiv 15$ mod 10 because the remainder of $15 \div 10$ and $5 \div 10$ are 5.
Example 2: What is a if $a \equiv 12$ mod 7?
Answer: $a = 5 + 7 * x$ with integer $x \geq 0$, e.g. 5, 12, 19, 26, 33, 40, etc.

Congruent Modulo

Congruent modulo equation

$$a \equiv b (\text{mod } n) \tag{6}$$

a and b are congruent modulo n if they share the same remainders when divided by n. (If its modulus on one side, its modulus on the other).

Example 1: $5 \equiv 15$ mod 10 because the remainder of $15 \div 10$ and $5 \div 10$ are 5.
Example 2: What is a if $a \equiv 12$ mod 7?
Answer: $a = 5 + 7 * x$ with integer $x \geq 0$, e.g. 5, 12, 19, 26, 33, 40, etc.

Equivalency vs Equality

$\equiv$ is not the same as $=$. In terms of modular arithmetic, $=$ represents the smallest solution to $\equiv$.

Bézout's identity [baizu]

Bézout's identity

$$ax + by = \gcd(a, b) \tag{7}$$

Bézout's identity is a theorem that proves two integers can be multiplied to equal the greatest common divisor (denominator is for fractions) of a and b.
x and y, known as Bézout's coefficients, can be computed by the extended Euclidean algorithm.
In the case of coprime integers a and b, $d = 1$.
Example: If $a = 12$, $b = 42$, then $d = \gcd(12, 42) = 6$. One possible solution to the theorem can be $(x, y) = (4, -1)$, so that
$4 \times 12 + -1 \times 42 = 6$.

Extended Euclidean Algorithm

Note: If mod n is on one side of an equation, it will also be on the other.
If a and m are coprime, the extended euclidean algorithm can be used to find b.
If we have time, I will dive into the math behind EEA, but I don't know how long this talk will take. PS It's also complicated.

Extended Euclidean Algorithm

$$ax + by = \gcd(a, b) \tag{8}$$

If a and m is known, the extended euclidean algorithm can be used to find b.
Example 1: If $a = 3$ and $b = 10$, what is x and y?

Totients

Euler's totient is the number of coprime integers less than n.
Coprime means "relatively prime" because a coprime integer may not actually be prime. Instead, it is prime to another number.
For example, 6 is not a prime number ($6 = 2 \times 3$), but it is **coprime** with 7 because $gcd(6, 7) = 1$.

Euler's Totient

For every prime p, $\phi(p) = p - 1$

Example 1: The totient of 13 is $13 - 1 = 12$ because 13 is a prime number.

Totients

Euler's totient is the number of coprime integers less than n.
Coprime means "relatively prime" because a coprime integer may not actually be prime. Instead, it is prime to another number.
For example, 6 is not a prime number ($6 = 2 \times 3$), but it is **coprime** with 7 because $gcd(6, 7) = 1$.

Euler's Totient

For every prime p, $\phi(p) = p - 1$

Example 1: The totient of 13 is $13 - 1 = 12$ because 13 is a prime number.
Example 2: What is Euler's totient of 2341 ($\phi(2341)$)?

Totients

Euler's totient is the number of coprime integers less than n.
Coprime means "relatively prime" because a coprime integer may not actually be prime. Instead, it is prime to another number.
For example, 6 is not a prime number ($6 = 2 \times 3$), but it is **coprime** with 7 because $gcd(6, 7) = 1$.

Euler's Totient

For every prime p, $\phi(p) = p - 1$

Example 1: The totient of 13 is $13 - 1 = 12$ because 13 is a prime number.
Example 2: What is Euler's totient of 2341 ($\phi(2341)$)?
Answer: $2341 - 1 = 2340$ because 2341 is a prime number.

Totients continued

What about the totients of nonprime numbers?
Because Euler's totient is a multiplicative function, the totients of nonprime numbers can be found.
If integer n is composed of two coprime integers, Euler's totient can be used.

Euler's Totient Multiplicative Property

For the totient of large, nonprime numbers, $\phi(m \times n) = \phi(m) \times \phi(n)$ if m and n are coprime.

Example 1: The totient of 77 ($\phi(77)$) can be factored into $\phi(7) \times \phi(11)$ because $gcd(7, 11) = 1$. 7 and 11 are both prime numbers, so $\phi(77) = (7 - 1) \times (11 - 1) = 60$.
Example 2: What is the totient of 15?

Totients continued

What about the totients of nonprime numbers?
Because Euler's totient is a multiplicative function, the totients of nonprime numbers can be found.
If integer n is composed of two coprime integers, Euler's totient can be used.

Euler's Totient Multiplicative Property

For the totient of large, nonprime numbers, $\phi(m \times n) = \phi(m) \times \phi(n)$ if m and n are coprime.

Example 1: The totient of 77 ($\phi(77)$) can be factored into $\phi(7) \times \phi(11)$ because $gcd(7, 11) = 1$. 7 and 11 are both prime numbers, so $\phi(77) = (7 - 1) \times (11 - 1) = 60$.
Example 2: What is the totient of 15? $\phi(15) = (5 - 1) \times (3 - 1) = 8$

Totients extra information

Final Totient theorem

$$\phi(m) = m(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})$$

Example 1: In order to find the totient of 20 ($\phi(20)$), split 20 into its coprime factors. $20 = 5 \times 2^2$ because $gcd(5, 2) = 1$. Although a coprime integer may be used multiple times (in the case of 2^2), only count the value once. Using the above theorem, $\phi(20) = 20(1 - \frac{1}{5})(1 - \frac{1}{2} = 8)$.
Note: $\phi(20) \neq \phi(10) \times \phi(2)$ because 10 and 2 are not coprime.

Totients extra information

Final Totient theorem

$$\phi(m) = m(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})$$

Example 1: In order to find the totient of 20 ($\phi(20)$), split 20 into its coprime factors. $20 = 5 \times 2^2$ because $gcd(5, 2) = 1$. Although a coprime integer may be used multiple times (in the case of 2^2), only count the value once. Using the above theorem, $\phi(20) = 20(1 - \frac{1}{5})(1 - \frac{1}{2} = 8)$.
Note: $\phi(20) \neq \phi(10) \times \phi(2)$ because 10 and 2 are not coprime.
Example 2: What is the totient of 568?

Totients extra information

Final Totient theorem

$$\phi(m) = m(1 - \frac{1}{p_1})(1 - \frac{1}{p_2})$$

Example 1: In order to find the totient of 20 ($\phi(20)$), split 20 into its coprime factors. $20 = 5 \times 2^2$ because $gcd(5, 2) = 1$. Although a coprime integer may be used multiple times (in the case of 2^2), only count the value once. Using the above theorem, $\phi(20) = 20(1 - \frac{1}{5})(1 - \frac{1}{2} = 8)$.
Note: $\phi(20) \neq \phi(10) \times \phi(2)$ because 10 and 2 are not coprime.
Example 2: What is the totient of 568?
Answer: 568 can be factorized into 71 and 8 (2^3), both coprime ($gcd(8, 71) = 1$). $\phi(568) = 568(1 - \frac{1}{71})(1 - \frac{1}{2}) = 280$.

Modulus

Modulus Formula

$$n = pq \tag{9}$$

n symbolizes the modulus in the RSA algorithm.
$\phi(\mathbf{n})$ generates the private key in the RSA algorithm.
n is the product of two primes, p and q.
Note: p and q are already coprime integers because both are prime anyway. $gcd(p, 1) = 1$, $gcd(q, 1) = 1$, and $gcd(p, q) = 1$.
Example: If p = 59 and q = 23, $n = 59 \times 23 = 1357$.

Modulus Totient

$$\phi(n) = (p - 1)(q - 1) \tag{10}$$

Because $n = pq$, $\phi(n) = \phi(p) \times \phi(q) = (p - 1)(q - 1)$

Private and Public Keys

Key Generation

$$de \equiv 1 \mod \phi(n) \tag{11}$$

d and **e** symbolize the private and public key in the RSA algorithm.
e is a user generated integer coprime with $\phi(n)$ that is greater than 1 and less than $\phi(n)$.
The public key e is used to generate the private key d using the Extended Euclidean algorithm, because d is the modular multiplicative inverse of e.
An interesting note: the RSA paper chooses d and generates e, optimizing decryption, but the modern implementations of RSA do the opposite.

Bézout's identity in modular inverse

Modular inverse

$$d \equiv e^{-1} \mod \phi(n) \tag{12}$$

Bézout's identity

$$ax + by = \gcd(a, b) \tag{13}$$

Assume e and $\phi(n)$ are coprime, making the $gcd(e, n) = 1$. So how can the Bézout's identity be applied to the modular inverse? It doesn't look like the standard form.

Well, subtracting 1 from both sides of the RSA formula yields: $de - 1 \equiv 0 \mod \phi(n)$.

And we know that any integer y for $y * \phi(n) \equiv 0 \mod \phi(n)$.

So, $x * e - 1 \equiv y * \phi(n) \mod \phi(n)$. Add 1 and subtract $\phi(n)$, and the final equation is $x * e - y * \phi(n) \equiv 1 \mod \phi(n)$, with x as d.

e and n are known, and we must find x and y in accordance to Bézout's identity using the Extended Euclidean Algorithm.

Background

Bob wants to send a secure message to Alice using RSA.
Alice sends her modulus n and her public exponent e.
Using those numbers, Bob can generate the ciphertext using $C \equiv M^e \mod n$ and send it to Alice.
Alice is able to decrypt the ciphertext using her private key using $M \equiv C^d \mod n$.
Meanwhile, Eve can only see n and e.

Qualitative Example

1. Alice generates two primes p and q.
2. Alice multiplies two factors p and q to create modulus n.
3. Alice solves $(p-1)(q-1)$ to generate the modulus totient $\phi(n)$.
4. Alice generates public exponent e so $1 < e < \phi(n)$ and coprime to $\phi(n)$.
5. Alice generates private exponent d so $d \equiv e^{-1} \mod \phi(n)$.
6. Alice sends n and e to Bob.
7. Bob generates plaintext m where $m < n$.
8. Bob computes ciphertext C by using plaintext m in $C \equiv m^e \mod n$.
9. Bob sends ciphertext C to Alice.
10. Alice decrypts ciphertext C by solving $m \equiv C^d \mod n$.

Real CTF examples

To solidify understanding of how RSA is meant to work, here are two examples of real CTF challenges.

Examples: rsa-pop-quiz, Relatively-Secure-Algorithm

Vulnerabilities

The vulnerabilities that I will be discussing are all human errors that result from a lack of understanding of RSA.
These vulnerabilities discussed will result in complete breaks of the RSA system, as these attacks are most important towards solving CTF challenges.

Small e

A small public key can render the RSA system entirely broken.
For developers rolling out a bootlegged RSA with no knowledge of how it works may choose speed over security and choose the public exponent to be $e = 1$. The issue with that is $C \equiv m^e \mod n$ would become $C = m \mod n$, and because $m < n$, $C = m$.

Example: Salty

Fun Fact

The most common choices for e are 3, 17, and 65537 because all are prime and all are Fermat numbers ($F_n = 2^{2^n} + 1$), making modular exponentiation faster.
65537 (F_4) is the largest Fermat prime discovered, since F_{11} is the largest Fermat number completely factored at 617 digits.

Small e and m

Given a small enough message and a small enough public exponent, the message can be easily reversed if $m^e < n$.
The formula $C = m^e \mod n$ can be reversed when $e = 3$ and the message is small because the message has not undergone modulus arithmetic.
Thus, $C^{1/e} = m$.

Example: mini-rsa, Modulus-Inutilis

Wiener's Attack

Wiener's attack relies on a small private exponent d to expose itself.

Wiener's Theorem

$$d < \frac{1}{3} N^{\frac{1}{4}} \tag{14}$$

$de \equiv 1 \mod \phi(n)$, meaning $de = k * \phi(n) + 1$, dividing $d * n$ means $\frac{e}{n} = \frac{k}{d} * \frac{\phi(n)}{n} + \frac{1}{d*n}$.
Find the convergents of the continued fraction expansion of $\frac{e}{n}$.

Example: dachshund, b00tl3gRSA2, sosig

Too many primes

Using more than 2 primes to compute the modulus lowers the security of RSA. It becomes easier to factor n, and therefore break the entire system.

Examples: b00tl3gRSA3, Manyprime

One prime

Generating large prime numbers can take a while on slow computers, and there are challenges about using only one prime.
The issue is the modulus totient will be exposed. Because n is a nonprime number, $\phi(n) = (p-1)(q-1)$. However, n as a prime number would make $\phi(n) = (n-1)$.

Example: Monoprime

Same primes

Instead of using one prime to save time, a challenge can involve using one prime twice.
Instead of $n = pq$, $n = p^2$. p can be determined by taking the square root of modulus n. Therefore, $\phi(n) = n(1 - \frac{1}{\phi(n)})$

Example: square-eyes

Conclusion

Any questions?

RSA Talk

Eagle

August 17, 2023