RSA cryptosystem in a nutshell

G. Averkov
Brandenburg University of Technology

May 8, 2024

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.
`https://doi.org/10.1145/359340.359342`

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.
`https://doi.org/10.1145/359340.359342`

Quote from Wikipedia:

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.
`https://doi.org/10.1145/359340.359342`

Quote from Wikipedia:

In April 1977, they spent Passover at the house of a student and drank a good deal of wine before returning to their homes at around midnight.

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.
`https://doi.org/10.1145/359340.359342`

Quote from Wikipedia:

In April 1977, they spent Passover at the house of a student and drank a good deal of wine before returning to their homes at around midnight. Rivest, unable to sleep, lay on the couch with a math textbook and started thinking about their one-way function.

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.
`https://doi.org/10.1145/359340.359342`

Quote from Wikipedia:

In April 1977, they spent Passover at the house of a student and drank a good deal of wine before returning to their homes at around midnight. Rivest, unable to sleep, lay on the couch with a math textbook and started thinking about their one-way function. He spent the rest of the night formalizing his idea, and he had much of the paper ready by daybreak.

The original 1978 paper of Rivest, Shamir and Adleman is just 7 pages long, written in a clear and non-technical language.
https://doi.org/10.1145/359340.359342

Quote from Wikipedia:

In April 1977, they spent Passover at the house of a student and drank a good deal of wine before returning to their homes at around midnight. Rivest, unable to sleep, lay on the couch with a math textbook and started thinking about their one-way function. He spent the rest of the night formalizing his idea, and he had much of the paper ready by daybreak. The algorithm is now known as RSA – the initials of their surnames in same order as their paper.

Alice

- chooses multi-digit prime numbers p and q with $p \neq q$, which she does not disclose to anyone.

Alice

- chooses multi-digit prime numbers p and q with $p \neq q$, which she does not disclose to anyone.
- calculates $n = pq$ and fixes an exponent $e \in \mathbb{N}$ such that $\gcd(e, (p-1)(q-1)) = 1$.

Alice

- chooses multi-digit prime numbers p and q with $p \neq q$, which she does not disclose to anyone.
- calculates $n = pq$ and fixes an exponent $e \in \mathbb{N}$ such that $\gcd(e, (p-1)(q-1)) = 1$.
- fixes (e, n) as a public key and makes it available to everyone, so that anybody can send her a plain text message $x \in \mathbb{Z}_n$ encrypted via the power function x^e.

Alice

- chooses multi-digit prime numbers p and q with $p \neq q$, which she does not disclose to anyone.
- calculates $n = pq$ and fixes an exponent $e \in \mathbb{N}$ such that $\gcd(e, (p-1)(q-1)) = 1$.
- fixes (e, n) as a public key and makes it available to everyone, so that anybody can send her a plain text message $x \in \mathbb{Z}_n$ encrypted via the power function x^e.
- calculates $d \in \mathbb{N}$ such that $de \equiv 1 \mod (p-1)(q-1)$ using the EEA.

Alice

- chooses multi-digit prime numbers p and q with $p \neq q$, which she does not disclose to anyone.
- calculates $n = pq$ and fixes an exponent $e \in \mathbb{N}$ such that $\gcd(e, (p-1)(q-1)) = 1$.
- fixes (e, n) as a public key and makes it available to everyone, so that anybody can send her a plain text message $x \in \mathbb{Z}_n$ encrypted via the power function x^e.
- calculates $d \in \mathbb{N}$ such that $de \equiv 1 \mod (p-1)(q-1)$ using the EEA.
- stores (d, n) as a private key in order to use the power function x^d on $\mathbb{Z}_n$ for decryption.

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Then, Alice

- receives $y = x^e$ from Bob

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Then, Alice

- receives $y = x^e$ from Bob
- calculates $y^d = (x^e)^d = x^{ed} = x$, which is Bob's plain text (we use Lemma 10, here)

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Then, Alice

- receives $y = x^e$ from Bob
- calculates $y^d = (x^e)^d = x^{ed} = x$, which is Bob's plain text (we use Lemma 10, here)
- and uses fast exponentiation to calculate y^d

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Then, Alice

- receives $y = x^e$ from Bob
- calculates $y^d = (x^e)^d = x^{ed} = x$, which is Bob's plain text (we use Lemma 10, here)
- and uses fast exponentiation to calculate y^d

Anytime Bob has a message $x \in \mathbb{Z}_n$ to be sent to Alice, he

- sends the plain text x to Alice as the cipher text $y = x^e$
- uses fast exponentiation to calculate x^e

Then, Alice

- receives $y = x^e$ from Bob
- calculates $y^d = (x^e)^d = x^{ed} = x$, which is Bob's plain text (we use Lemma 10, here)
- and uses fast exponentiation to calculate y^d

Eve

- may attempt to break the cryptosystem by trying to determine the private key from the public one.

Eve

- may attempt to break the cryptosystem by trying to determine the private key from the public one.
- knows that if she gets p and q by factorizing $n = pq$ into prime factors, she can do the same computations that Alice did to generate the private key.

Eve

- may attempt to break the cryptosystem by trying to determine the private key from the public one.
- knows that if she gets p and q by factorizing $n = pq$ into prime factors, she can do the same computations that Alice did to generate the private key.
- does not know any elaborate way to factorize integers into primes. If the smaller of the two primes p and q has k binary digits, then by just trying consecutively starting from 3 all possible odd integers as possible divisors of n, would determine the smaller prime within about 2^k iterations. If k is something like 200, such an approach is by now means tractable.

$$\begin{aligned} 2^{200} =& 1606938044258990275541962092 34 \\ & 1162602522202993782792835301376 \end{aligned}$$

With trillion iterations in a second, one would break the cryptosystem in about $3 \cdot 10^{40}$ years.

Eve

- can also try to to break a particular cipher text y she intercepts by looking for an appropriate x that satisfies $x^e = y$.

Eve

- can also try to to break a particular cipher text y she intercepts by looking for an appropriate x that satisfies $x^e = y$.
- does not really have any idea of solving the equation $x^e = y$ in x than just trying all elements of $\mathbb{Z}_n$ as possible values of x.

Eve

- can also try to to break a particular cipher text y she intercepts by looking for an appropriate x that satisfies $x^e = y$.
- does not really have any idea of solving the equation $x^e = y$ in x than just trying all elements of $\mathbb{Z}_n$ as possible values of x.
- runs into this tractability problem as with factorization of n into prime factors.

Eve may also try using

Eve may also try using

- smarter algorithms for integer factorization (there are some that are faster, but the are still exponential-time in the bit size of n)

Eve may also try using

- smarter algorithms for integer factorization (there are some that are faster, but the are still exponential-time in the bit size of n)
- a quantum computer to factor integer in polynomial time in the bit size of n, but quantum computers that could do this and have a large enough quantum memory do not seem to be available yet.

Eve may also try using

- smarter algorithms for integer factorization (there are some that are faster, but the are still exponential-time in the bit size of n)
- a quantum computer to factor integer in polynomial time in the bit size of n, but quantum computers that could do this and have a large enough quantum memory do not seem to be available yet.
- "dirty tricks" (aka side-channel attack)

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done
- Fast exponentiation for y^d does one iteration for each binary digit of d.

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done
- Fast exponentiation for y^d does one iteration for each binary digit of d.
- When the binary digit is 1, there is squaring and multiplication

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done
- Fast exponentiation for y^d does one iteration for each binary digit of d.
- When the binary digit is 1, there is squaring and multiplication
- When the binary digit is 0, there is only squaring.

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done
- Fast exponentiation for y^d does one iteration for each binary digit of d.
- When the binary digit is 1, there is squaring and multiplication
- When the binary digit is 0, there is only squaring.
- So, looking at the power profile, she detects the binary digits of d.

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done
- Fast exponentiation for y^d does one iteration for each binary digit of d.
- When the binary digit is 1, there is squaring and multiplication
- When the binary digit is 0, there is only squaring.
- So, looking at the power profile, she detects the binary digits of d.

Side channel attack. Eve, being a hacker, does not want to rack her brain over math and instead does this:

- uses a spy process to track the power consumption of the device, on which deciphering is done
- Fast exponentiation for y^d does one iteration for each binary digit of d.
- When the binary digit is 1, there is squaring and multiplication
- When the binary digit is 0, there is only squaring.
- So, looking at the power profile, she detects the binary digits of d.

Alice can fend off this attack by making the implementation of fast exponentiation safe.

Are there enough primes?

- Prime number theorem tells, roughly, that the probability of a random k-digit number to be prime is proportional to $1/k$.

Are there enough primes?

- Prime number theorem tells, roughly, that the probability of a random k-digit number to be prime is proportional to $1/k$.
- This means that, starting with a random odd number and then successively increasing this number by 2 to get to the next odd number, one would get to a prime number in a reasonable amount of time.

Are there enough primes?

- Prime number theorem tells, roughly, that the probability of a random k-digit number to be prime is proportional to $1/k$.
- This means that, starting with a random odd number and then successively increasing this number by 2 to get to the next odd number, one would get to a prime number in a reasonable amount of time.

Are there enough primes?

- Prime number theorem tells, roughly, that the probability of a random k-digit number to be prime is proportional to $1/k$.
- This means that, starting with a random odd number and then successively increasing this number by 2 to get to the next odd number, one would get to a prime number in a reasonable amount of time.

Can one test a number for primality quickly?

- Yes, there is a primality test, whose running time is polynomial in the bit size of the number in the input. Manindra Agrawal, Neeraj Kayal and Nitin Saxena from Indian Institut of Technology Kanpur. Published in 2002.

Are there enough primes?

- Prime number theorem tells, roughly, that the probability of a random k-digit number to be prime is proportional to $1/k$.
- This means that, starting with a random odd number and then successively increasing this number by 2 to get to the next odd number, one would get to a prime number in a reasonable amount of time.

Can one test a number for primality quickly?

- Yes, there is a primality test, whose running time is polynomial in the bit size of the number in the input. Manindra Agrawal, Neeraj Kayal and Nitin Saxena from Indian Institut of Technology Kanpur. Published in 2002.
- In practice, one usually employs a probabilistic test that can generate false positives. That is, if the number is discarded as composite, it is in fact composite. However, if the number is accepted as prime by the test, it is quite likely a prime but not for sure.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.
- We leave out the formal definition.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.
- We leave out the formal definition.
- Existence of one-way function is not proven yet (hard open problem).

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.
- We leave out the formal definition.
- Existence of one-way function is not proven yet (hard open problem).
- However, practically, we just want to make sure that the following is the case.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.
- We leave out the formal definition.
- Existence of one-way function is not proven yet (hard open problem).
- However, practically, we just want to make sure that the following is the case.
- We know how to calculate $f(x)$ for a given x quickly

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.
- We leave out the formal definition.
- Existence of one-way function is not proven yet (hard open problem).
- However, practically, we just want to make sure that the following is the case.
- We know how to calculate $f(x)$ for a given x quickly
- No one is yet capable to solve the equation $f(x) = y$ for the unknown x efficiently.

One-way function:

- Roughly speaking, a function that is easy to evaluate but hard to invert.
- Calculating $f(x)$ for given x – easy.
- Solving the equation $f(x) = y$ in unknown x – hard.
- We leave out the formal definition.
- Existence of one-way function is not proven yet (hard open problem).
- However, practically, we just want to make sure that the following is the case.
- We know how to calculate $f(x)$ for a given x quickly
- No one is yet capable to solve the equation $f(x) = y$ for the unknown x efficiently.
- In this respect, x^e from the RSA is a “one-way function from the practical perspective”

Alice can sign a message $x \in \mathbb{Z}_n$.

- Calculate $s = x^d$, the so-called signature, and send it along with x.

Alice can sign a message $x \in \mathbb{Z}_n$.

- Calculate $s = x^d$, the so-called signature, and send it along with x.

Alice can sign a message $x \in \mathbb{Z}_n$.

- Calculate $s = x^d$, the so-called signature, and send it along with x.

Bob can verify that x is a message from Alice as follows:

- Calculate $s^e = (x^d)^e = x$ and check that the message x comes out.

In order to fake the x signature, Eve

- could either determine d, which is infeasible, as was described above
- or try to solve the equation $s^e = x$ for the unknown signature s, which is infeasible, too.

Some practical aspects, which we don't want to address in detail:

- We presented RSA from the textbook. It contains the main idea, but in practice adjustments are made and details should be clarified.

Some practical aspects, which we don't want to address in detail:

- We presented RSA from the textbook. It contains the main idea, but in practice adjustments are made and details should be clarified.
- In order to convert any kind of data (texts etc.) into a number, one can convert any kind of symbols into bit sequences (ASCII encoding etc.) and then view a bit string as an integer written in the binary.

Some practical aspects, which we don't want to address in detail:

- We presented RSA from the textbook. It contains the main idea, but in practice adjustments are made and details should be clarified.
- In order to convert any kind of data (texts etc.) into a number, one can convert any kind of symbols into bit sequences (ASCII encoding etc.) and then view a bit string as an integer written in the binary.
- Such an integer determines a coset modulo n.

Some practical aspects, which we don't want to address in detail:

- We presented RSA from the textbook. It contains the main idea, but in practice adjustments are made and details should be clarified.
- In order to convert any kind of data (texts etc.) into a number, one can convert any kind of symbols into bit sequences (ASCII encoding etc.) and then view a bit string as an integer written in the binary.
- Such an integer determines a coset modulo n.
- If the bit-strings are two long, then one could split them into blocks. In practice, it is important to make such a conversion properly to make the encryption safe (padding etc.)

Some practical aspects, which we don't want to address in detail:

- We presented RSA from the textbook. It contains the main idea, but in practice adjustments are made and details should be clarified.
- In order to convert any kind of data (texts etc.) into a number, one can convert any kind of symbols into bit sequences (ASCII encoding etc.) and then view a bit string as an integer written in the binary.
- Such an integer determines a coset modulo n.
- If the bit-strings are two long, then one could split them into blocks. In practice, it is important to make such a conversion properly to make the encryption safe (padding etc.)
- When one signs (long) messages, one signs not the original text x, but rather the hash value $h(x)$ of the text x, where h is a so-called hash function.