Online Cryptography

Extended Essay

Subject: Mathematics

Word Count: 3,584

18 November 2022

Table of Contents

What Is RSA encryption, and how is it still important for privacy in today's world?

Introduction

Anytime you connect to a website, connect to the internet, search the web, or even send a text message, your electronic device utilizes cryptography to maintain privacy and online safety. Typical cryptographic messages code the message you send to somebody, and that person then deciphers your message to reveal the contents. This prevents a possible data breach, where unwanted people can view your web searches, phone calls, or even bank transactions. As supported by Cornell University, communication falls under two major types of ciphers: the traditional symmetric-key cryptosystem and the public-key cryptosystem. Let's say Alice and Bob want to communicate using the conventional symmetric-key cryptosystem. In this model, they must agree to a set key before sending messages. That way, Alice and Bob can communicate without eavesdroppers or unwanted listeners (“Public Key Cryptography II”). According to Rhodes, while this approach makes sense, it encounters two significant problems in its logic. The first is the private key exchange problem. This is where Alice and Bob are located geographically far apart, so they cannot agree to a private key in person. Since their only possible means of communication are messaging each other, they would have to share their private key with each other over the internet. This situation is called the key distribution paradox. Once their private key is exposed to the internet, the messages can be decrypted by anyone, thus nulling the idea of encrypting the messages in the first place. The second major problem is the sender verification problem: if one were to intercept a private key, one could also use the key to secretly encrypt messages. In this case, anyone could pretend to be the original sender, and the receiver would have no way to confirm the identity of the sender (“RSA Algorithm Overview”). Cornell University recognizes that these problems puzzled engineers for years until, eventually, in the

'70s, they came up with something called public-key encryption to get around the private key exchange problem and the sender verification problem that is found in symmetric-key encryption. In the public-key cryptosystem, Alice and Bob communicate without knowing a set key. Alice creates a public key and puts it out into the world. But Alice also has a secret private key, which they keep only to themselves. Alice's public key is essentially a padlock, and the key to this padlock is her private key. Let's say Alice gave this padlock to Bob. Bob may then write his message in a box and then put the padlock on the box and lock it. Since Alice is the only one with the private key, she is able to extrapolate the box's contents in private ("Public Key Cryptography II"). This approach nulls the idea of sharing an agreed-upon private key, solving the private key exchange problem. The public-key cryptosystem also signs the message with a digital certificate, confirming who the messenger is, therefore solving the sender verification problem and confirming the key pair owner as a trustworthy source (Encryption, "What is RSA?"). The public-key approach is the fundamental basis for the day-to-day encryption that surrounds our world and protects online privacy. While there have been different types of public-key systems, none are as widely used as the RSA Cipher. The RSA Cipher was developed by Rivest Shamir and Leanard Adelman in 1977, and it uses prime numbers in order to achieve data encryption using the public-key model (Grifski, "Number Theory Behind RSA"). RSA is crucial for privacy in today's world since nearly everything sent over the internet uses the RSA cipher due to its security and reliability.

What is RSA and how does it work?

"RSA derives its security from the difficulty of factoring large integers that are the product of two extremely large prime numbers" (Guest, "Diffie-Hellman and RSA"). This works

through a type of mathematics called modular arithmetic; Take Alice and Bob, for example (see table 1).

Table 1

Modular Arithmetic steps for RSA

Step Number	Process
1	Alice first chooses two different prime numbers, P and Q, which she keeps secret. P and Q are about 100 digits long each.
2	Alice calculates her *modulus* N by multiplying P and Q together $(N = P * Q)$.
3	Alice then calculates Z by multiplying $(P - 1)$ and $(Q - 1)$ together. $Z = (P - 1) * (Q - 1)$. At this point Alice no longer has to know what P and Q are.
4	Alice will then choose an *encryption exponent* E. $(1 < E < Z)$ so E and Z have absolutely no factors in common.
5	Alice then finds her *decryption exponent* D by solving for the number D, such that the product of E and D is 1 *modulo* Z. $1 \ mod \ Z = ED$. This can be done using the Euclidean algorithm on a computer.
6	Alices then shares the pair of numbers E and N (these make up the public-key), while she keeps D secret (this makes her private key).

7	If Bob wants to send a message to Alice, he must convert his message into a number M. (He can do this using the ASC11 code (American Standard Code for Information Interchange)).
8	Then Bob must calculate $C = M^E (mod\ N)$, and he sends C to Alice.
9	When Alice receives C, she will compute $C^D (mod\ N) = M^{ED} (mod\ N)$. Since E and D were specially calculated using the large primes, this equation will give Alice M. Thus, Alice gets Bobs message.

Source: Cornell University. "Primes, Modular Arithmetic and Public Key Cryptography II." n.d., pi.math.cornell.edu/~mec/2003-2004/cryptography/rsa/rsa.html. Accessed 12 April 2022.

It is true that Alice can compute $C^D (mod\ N) = M^{ED} (mod\ N)$ to find M; however, there is another set of mathematics that proves $M^{ED} (mod\ N) = M$ that is separate from the steps taken in modular arithmetic. The process is called Fermat's Little Theorem and Euler's theorem. Fermat's Little Theorem states that "For any prime number p and any number a with $a < p,\ a^p (mod\ p) = a$" (Grifski, "Number Theory Behind RSA"). Euler's theorem proved that "If $n = p * q$ is the product of two primes, and a is any number such that $a < n$, then $a^{((p-1)(q-1)) +1} (mod\ n) = a$" (Katz, "RSA Encryption"). In layman's terms, this means that the exponent of a cancels out mod n, yielding out a. In the case listed in Table 1, the exponent of M, ED, cancels out mod N, thus yielding M. The reason this is so secure is that the entire process of RSA is based on enormous primes, which are easy to multiply and form equations with; however, they are exceedingly difficult to factor.

Bit Vulnerability in RSA

Factoring plays a prominent role in the security of RSA, mainly due to the sheer amount of computational power that is required to factor large primes (Katz, “RSA Encryption”). The public key must be decrypted by the private key, and this can be derived from N (Lake, “How RSA Works”). If somebody can calculate N, then the entire system falls apart. While this may seem like a massive vulnerability, the process of calculating N without knowing P and Q is almost impossible with modern technology. Thus, in turn, RSA is currently highly secure because of the inability to factor large primes. However, that doesn't mean it hasn't been accomplished. The lower the RSA key's data size, the easier it is to brute factor N to derive the private key (Katz, “RSA Encryption”). Data sizes range from 512 bit, 768 bit, 2048 bit, and 4096 bit. Currently, the most secure RSA key is 4096 bit, which is 1234 digits long. Because of this extremely high key length, the computational power required to brute factor is unfeasible (Encryption, “What is RSA?”). According to a research team led by Thorsten Kleinjung who successfully brute-forced a 768 key, "Factoring a 1024-bit RSA modulus would be about a thousand times harder, and a 768-bit RSA modulus is several thousand times harder than a 512-bit one.” Despite demonstrating that a fault-based attack on the RSA algorithm is possible," it also must be commonly understood that it costs around $76,000 to purchase the equipment and computational power necessary (Kleinjung, 1). The rise of computational power and computer capability pose a threat; however, we are far from being able to factor a 1024-bit RSA modulus.

Quantum Computing Vulnerability in RSA

Even so, the rapid research and development of Quantum Computers raise questions regarding the security of RSA. A quantum computer is a computer that uses quantum mechanical phenomena to perform calculations. Quantum computers are able to store and process

information using quantum bits or qubits. These computers are different in many ways from the computers that are in use today. For example, a quantum computer can be in multiple states simultaneously, whereas a classical computer can only be in one state at a time. This allows quantum computers to perform several calculations at once. In theory, a quantum computer could use its power to factor in large numbers, which is the basis of RSA encryption. This would be done using Shor's algorithm. Shor's algorithm is a quantum algorithm for factoring large integers that was devised by Peter Shor in 1994. It is the most efficient known classical algorithm for factoring large integers, with a runtime of polynomial time. Since Shor introduced the concept of quantum computing in calculating factorization, they have been rapidly increasing in power. However, it is important to note that in order to factor a 2048-bit RSA key, a quantum computer would require a total of 20 million qubits to reliably crack the key. In today's breakthrough in quantum computing, we are only able to use 70 qubits. Thus, society is ways away from developing a quantum computer capable enough to run Shor's algorithm on a 2048-bit key; however, this vulnerability is important to keep in mind when considering sensitive information at stake in the future, such as military records or confidential government information.

RSA inefficiencies, and the solution; AES

Bit size and Quantum computing are not the only concerns for RSA, despite its strong stance on security. Since RSA is such an extensive algorithm, it takes enormous amounts of time to generate the actual keys and perform calculations with enormous primes (Upadhyay, “RSA and Large Files”). This is an issue considering the typical applications of RSA. If people are connecting to a web page, they don't want to wait for a web page to load minute after minute or for a credit card transaction at the gas station or grocery store to take longer than a few seconds. Even regarding phone calls and text messages, the world is built on quick internet speed, so it

wouldn't make sense for these applications to take forever to load. RSA is too inefficient to use for very long data, as the RSA key must be just as large as what it is encrypting (Rhodes, “RSA Algorithm Overview”). So for a 4096-bit RSA key, the maximum message length you could encrypt is 4096 bits. For reference, this sentence is approximately 60 bites long. With encrypting entire websites, phone calls, or even text messages, the data size would be too big to process, and the encryption itself would take too long to complete if it can even complete it with your device's computational power (Upadhyay, “RSA and Large Files”). Luckily, there is a workaround to this issue that still establishes RSA's importance to privacy in the modern world. The gold standard symmetric key encryption: AES. According to Crawford, AES stands for Advanced Encryption Standard, and it utilizes the same key to encrypt and decrypt data. Typically AES is used to secure data since it requires less computational power than RSA and is much faster. Because AES is a symmetric cipher, it can bulk encrypt larger amounts of data, while asymmetric ciphers like RSA can only encrypt small amounts of data sufficiently (“AES Encryption”). As previously discussed, symmetric key ciphers are unreliable in key exchange because a key cannot be securely sent. The solution to this was asymmetric encryption, RSA, which creates two keys from prime numbers that can decrypt data derived from the other key. Due to the fact that AES is better suited for quick, secure data transfer, and RSA is better suited for key exchange, modern web encryption has evolved to use a combination of both (Upadhyay, “RSA and Large Files”). This conjunction between AES and RSA encryption would ultimately fail if RSA weren't as powerful at key exchange as it currently is. Thus, even in a combined solution for secure data transmission, RSA is still the prevalent factor in implementing privacy.

Alternatives to RSA

However, RSA's utilization as the backbone for modern privacy could change as technology further develops. AES has proved to be a solution to RSA's inability to encrypt large amounts of data, but eventually, technology will develop to the point where it can crack a 1024-bit key used for RSA (Crawford, “AES Encryption”). Then eventually, computers will be able to crack a 2048-bit, a 4096-bit, and so forth (Kleinjung, 1). Generally, this wouldn't be an issue as the technology and money required for brute forcing a 1024-bit key is unfeasible, but it is possible for those who can afford it. Or as Goodin states, organizations with a substantial billion-dollar budget for "groundbreaking cryptanalytic capabilities." Otherwise known as the NSA. The NSA; or National Security Agency, records network traffic and invests billions of dollars annually into managing data ranging from global leaders to 20% of the top million websites on the internet (“How the NSA can Break RSA”). The problem is that recorded traffic from 20 years ago, which used 512 or 768 keys, could be decrypted using brute force methods that are possible today, with more powerful computers to perform more complex calculations. This means that the NSA or other giant data management corporations can go back and decrypt information that was protected by exploiting the key with modern technology (Matas, “NSA Programs by Snowden”). Everything on the internet currently encrypted today will be at risk of exposure to decryption as computer development increases, creating a world where private information can no longer be protected by the current conjunction of RSA and AES (Goodin, “How the NSA can Break RSA”). Thus, society needs to utilize new methods of encryption that will surpass the security of RSA in order to prevent the downfall of privacy on the internet. The best alternatives to RSA are ciphers that use Forward Secrecy, which is not present in RSA. In RSA, the private key is the one vulnerability point of the cipher. According to an article by Namecheap, this is because a server may create the public and private keys for several different

communication sessions, meaning that the security for these sessions relies on the secrecy of the private key. If the private key is somehow leaked or compromised, people, such as the NSA, can go back to previously recorded traffic and decrypt information using the long-term public and private key pair, resulting in sensitive data being stolen. It is as if somebody were to break into your apartment, he would be able to see everything that it currently contains. Not only that, but everything that it used to contain. Every conversation, every person who walked in, and every meal cooked. With Forward Secrecy "even though someone breaks into a house, he still won't know what was happening before he got there." Forward Secrecy works by having unique key parameters for every communication session and erasing that key once the session has been completed. This means the session won't rely upon long-term keys that can be recorded and cracked with newer computers in the future ("Perfect Forward Secrecy").

Diffie-Hellman

Forward Secrecy is provided through the idea of the Diffie-Hellman key exchange, which uses similar mathematics to RSA. Take Alice and Bob as an example again (see table 2).

Table 2

Modular Arithmetic steps for Diffie-Hellman

Step Number	Process
1	Alice and Bob publicly agree on a prime number P, and a base number N.
2	Alice chooses a number A, called her *secret exponent*. Bob also chooses his *secret exponent* B. (Both A and B should be relatively prime to N, since A should have no common factors with N, and neither should B.

3	Alice then computes $J = N^A (mod\ P)$. She then sends J to Bob.
4	Bob computes $K = N^B (mod\ P)$ and sends K to Alice.
5	Alice now takes K, and computes $K^A (mod\ P)$. Bob does the same thing, computing $J^B (mod\ P)$. They both get the same number after computing this. This is because both of these expressions equal $N^{AB} (mod\ P)$, without exposing each other's *secret exponents*.

Source: Cornell University. "Primes, Modular Arithmetic and Public Key Cryptography II." n.d., pi.math.cornell.edu/~mec/2003-2004/cryptography/rsa/rsa.html. Accessed 12 April 2022.

On its own, Diffie-Hellman cannot provide Forward Secrecy or even a fully secure key exchange. Despite the mathematics being similar to RSA, Diffie-Hellman can't encrypt messages or digitally sign them. Diffie-Hellman is used more for creating an initial session key, while other more secure algorithms use that key to start a session (Encryption, “What is RSA?”). For Forward Secrecy, Diffie-Hellman is utilized to create an Ephemeral key. "Ephemeral keys are temporary keys used for one instance of a protocol execution and then thrown away" (Walton, “Certificate and Public Key Pinning”). The Diffie-Hellman Ephemeral, DHE for short, provides Forward Secrecy; however, it lacks security compared to RSA. A more secure cipher must be used with DHE to achieve Forward Secrecy and high security. For this, the upcoming successor of RSA must be utilized: Elliptic Curve Cryptography.

Elliptic Curve Cryptography

Elliptic Curve Cryptography, ECC, provides the same capabilities as RSA but with the bonus of faster key generation, which is where RSA lacked. According to Svetlin Nakov's work in progress Ebook on "Practical Cryptography for Developers," ECC is based on elliptic curves over finite fields and utilizes a discrete logarithm problem to maintain security. In ECC, the private key is composed of integers in the range of the curve's field size, while the public key is two random integers {x,y} laying on the curve. ("Elliptic Curve Cryptography"). Elliptic curves can be defined in an equation such as "$y^2 = x^3 + ax + b$" (see Fig. 1).

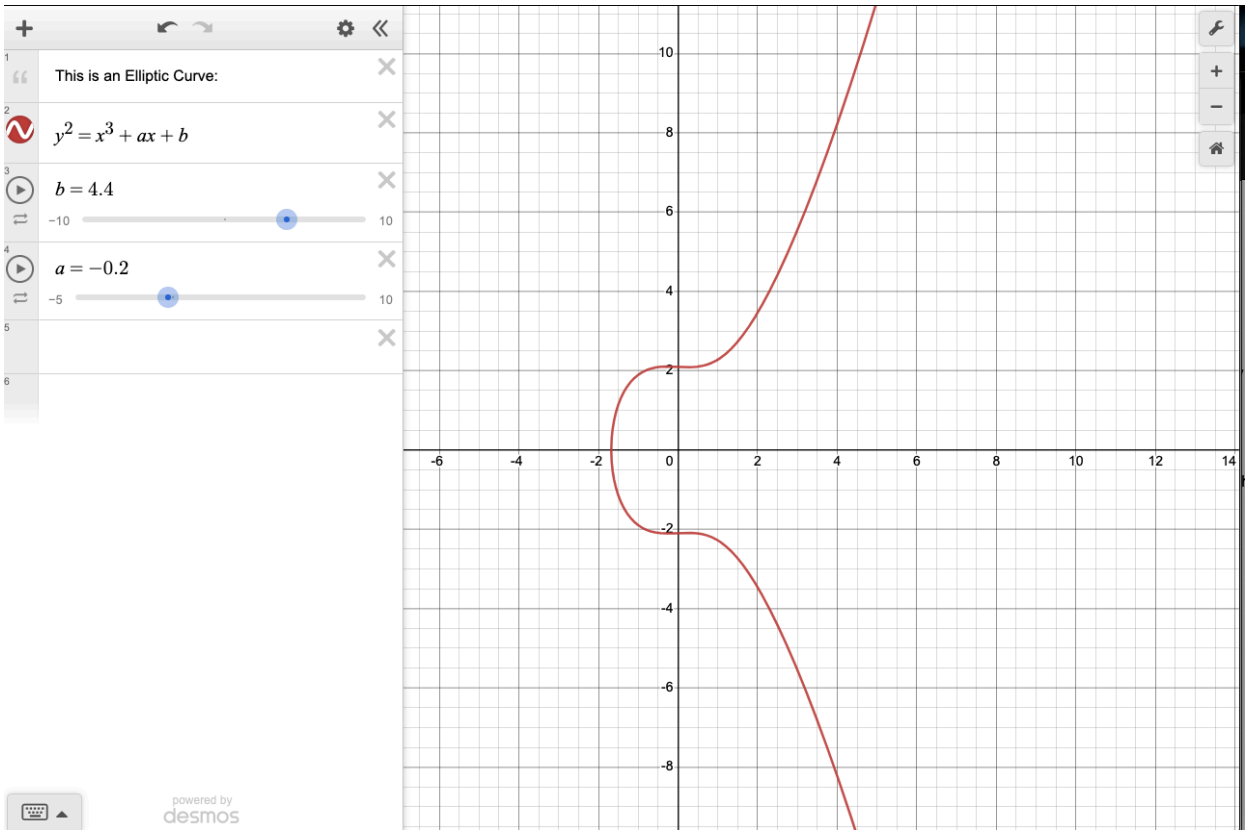

Fig. 1. Elliptic Curve calculated using online graphing software "Desmos"

ECC on its own is not enough to fully compete with RSA, as it also lacks Forward Secrecy. Thus, this cipher is combined with DHE, where DHE creates the temporary key, and ECC is

used for the encryption using said key. The combination of ECC and DHE became known as ECDHE, which allows two parties to use an elliptic curve key pair to establish communications on an insecure channel (Nakov, “Elliptic Curve Cryptography"). Not only does it accomplish Forward Secrecy, which RSA lacks, but it has extremely high security and fast key exchange rates (Namecheap, “Perfect Forward Secrecy”). ECDHE mitigates all the possible vulnerabilities encountered in RSA thus far. As humanity becomes more technologically advanced, encryption methods should also evolve to maintain privacy in the modern world.

Analysis

Overall, RSA is still an extremely prevalent form of encryption and will likely continue to be so in the upcoming years. RSA relies heavily on modular arithmetic, a form of mathematics often associated with programming, and uses the product of two large primes to asymmetrically create key pairs, allowing two parties to communicate in private. Even though this process is prolonged because the cipher requires heavy amounts of computational power for larger transmissions, RSA is still the backbone of modern encryption, as RSA is used for key exchange in other ciphers, specifically AES. This is because AES solves one of the reliability problems concerning RSA. The bit size for data encrypted with RSA is equal to the size of the data itself. This means that for extensive amounts of data, a computer will have to calculate the RSA key using large amounts of computational power, making the process extremely slow. This is a concern for practicality as well, as credit cards or phone calls are encrypted with RSA. Loading speeds for a website or simple credit card transactions can take much longer than expected, making RSA encryption inefficient for general encryption. As a workaround to this problem, AES is used to encrypt the data itself, while RSA is used solely for key exchange between the sender and receiver. However, another concern for RSA is the rise in computational complexity,

which puts RSA at risk due to the possibility of brute forcing a key. In order to prevent individuals from looking back at recorded traffic and brute forcing an RSA key with modern technology as society progresses into the future, different methods must be implemented in online security. Similar mathematics using modular processes introduced Diffie-Hellman key exchange, which was used to develop Diffie-Hellman Ephemeral, which introduced Forward Secrecy in cryptography. RSA lacks Forward Secrecy, as it uses lasting keys for different communication sessions, putting information in danger of being decrypted at a later date. Diffie-Hellman Ephemeral, DHE, temporarily creates a key and deletes it to null this issue. When combined with Elliptical Curve Cryptography, ECC, DHE gets the security RSA has without the vulnerabilities. Researching this has proven insightful into why security on the internet actually works and how far privacy can go within the world. This research has also shown that privacy is in danger with the development of technology and that society must implement new ways for encryption on a global level.

Conclusion

RSA encryption is a viable solution for online privacy for the time being; however, society should move towards ECDHE to avoid exposed data in the future. This means replacing the standard of RSA+AES encryption over international servers, web browsers, phone services, banks, and anything transmitted over the internet. RSA is still extremely prevalent and important in today's world for privacy because of its security and reliability as a cipher, but as technological power evolves, RSA might not be so prevalent in tomorrow's world.

Works Cited

Cornell University. "Primes, Modular Arithmetic and Public Key Cryptography II." n.d., pi.math.cornell.edu/~mec/2003-2004/cryptography/rsa/rsa.html. Accessed 12 April 2022.

Crawford, Douglas. "AES Encryption | Everything You Need to Know about AES." n.d., proprivacy.com/guides/aes-encryption. Accessed 09 September 2022.

Encryption Consulting. "RSA | what is RSA? | Encryption Consulting." *Encryption Consulting | Encryption Consulting,* n.d., 2020, www.encryptionconsulting.com/education-center/what-is-rsa/. Accessed 12 April 2022.

Grifski Jeremy. "Understanding the Number Theory Behind RSA Encryption | the Renegade Coder." *The Renegade Coder,* n.d., 2019, therenegadecoder.com/code/understanding-the-number-theory-behind-rsa-encryption/. Accessed 12 April 2022.

Goodin, D. "How the NSA Can Break Trillions of Encrypted Web and VPN Connections". Ars Technica, n.d., https://arstechnica.com/information-technology/2015/10/how-the-nsa-can-break-trillions-of-encrypted-web-and-vpn-connections/. Accessed 08 September 2022.

Guest Blogger: Anastasios Arampatzis. “Diffie-Hellman Key Exchange vs. RSA Encryption| Venafi.” Venafi.com, 2019, www.venafi.com/blog/how-diffie-hellman-key-exchange-different-rsa.

Help Net Security. "RSA Authentication Weakness Discovered - Help Net Security." *Help Net Security,* n.d., 2010, www.helpnetsecurity.com/2010/03/04/rsa-authentication-weakness-discovered/. Accessed 12 April 2022.

Katz Alexander, Ng Aloysius, and Bourg Patrick. "RSA Encryption | Brilliant Math & Science Wiki." n.d., brilliant.org/wiki/rsa-encryption/. Accessed 12 April 2022.

Kleinjung Thorsten. *Factorization of a 768-bit RSA Modulus*. 2010.

Lake Josh. "What is RSA Encryption and How Does it Work?." *Comparitech,* 2018, www.comparitech.com/blog/information-security/rsa-encryption/. Accessed 12 April 2022.

Ma, D. "RSA All Math Considered". *All Math Considered*, n.d., https://allmathconsidered.wordpress.com/tag/rsa/. Accessed 12 April 2022.

Matas. "Summary of the NSA Programs Leaked by Snowden". Cogipas.com, 2015, https://www.cogipas.com/snowden-leaks-summary-of-nsa-programs/. Accessed 08 September 2022.

Mayo Sherry. "Some Mathematical Ideas from Modular Arithmetic Used in RSA." n.d., cd.textfiles.com/group42/crypto/pgp/modulus.htm. Accessed 12 April 2022.

Nakov, Svetlin. "Elliptic Curve Cryptography (ECC) - Practical Cryptography for Developers." n.d., https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc. Accessed 10 September 2022.

Namecheap. "Perfect Forward Secrecy. What is it Is? - SSL Certificates - Namecheap.com." n.d., https://www.namecheap.com/support/knowledgebase/article.aspx/9652/38/perfect-forward-secrecy-what-it-is/. Accessed 11 September 2022.

Rhodes Delton. "What is RSA Encryption? an Overview of the RSA Algorithm." *Komodo Academy | En,* n.d., 2020, komodoplatform.com/en/academy/rsa-encryption/. Accessed 12 April 2022.

Upadhyay, S. "Why RSA is NOT Used to Encrypt LARGE Files?" Medium, 2021, https://infosecwriteups.com/why-rsa-is-not-used-to-encrypt-large-files-d3172d83febd. Accessed 08 September 2022.

Walton, Jeffery, J. Steven, J. Manico, K. Wall, and R. Iramar. "Certificate and Public Key Pinning | OWASP Foundation." n.d., https://owasp.org/www-community/controls/certificate_and_public_key_pinning. Accessed 11 September 2022.