SUNY POLYTECHNIC INSTITUTE

CS 532: Cryptography and Data Security

RSA Cryptography:

Foundations, Security Issues, Real-World Failures, and Modern Challenges

Taylor Hunter
Kelly Powell

December 2025

Table of Contents

Contents

Abstract

RSA is one of the central public-key systems used in modern cryptography and continues to play an important role in secure communication. Its security comes from the use of modular arithmetic and from the difficulty of factoring large composite numbers. These properties allow RSA to support major applications such as TLS, certificates, software signing, and authentication. Although RSA has been successful for many years, practical use has also shown that its security depends on correct implementation. Problems such as weak random number generation, poor key construction, insecure padding, and side-channel leakage have resulted in several well-known failures.

This report provides a clear overview of RSA, beginning with the basic mathematics of key generation and operations. It then examines how RSA is used for digital signatures and how it fits within protocols such as TLS. The paper also reviews important incidents where RSA implementations failed and explains why these failures occurred. A hands-on demonstration with a deliberately weak RSA system is included to show how insecure parameters allow an attacker to recover private keys. The report concludes with a discussion of quantum computing and the need for future transitions to post-quantum methods. The goal of this work is to present an organized and complete review of RSA that connects theory with practice and explains the main ideas, issues, and long-term considerations.

1 Introduction

RSA cryptography represents a landmark achievement in digital security. Introduced in 1978, RSA became the first practical public-key encryption system capable of supporting confidentiality and authentication on a large scale. Before RSA, secure communication relied entirely on symmetric-key cryptography, which requires both parties to share a secret key. As global networks expanded, this model became increasingly impractical due to the difficulty of distributing keys securely.

RSA revolutionized this model by introducing asymmetric key pairs—one public key for encryption and one private key for decryption. This allowed secure communication without prior interaction and laid the foundation for secure web browsing, digital certificates, VPN authentication, secure email, and software integrity verification.

This expanded report retains all original content while introducing new sections that explore RSA's broader cryptographic role, advanced protocol integration, real-world failures, and post-quantum challenges.

2 Background and Cryptographic Context

Before the invention of public-key cryptography, all secure communication depended on symmetric-key systems. These systems provide confidentiality but suffer from a major limitation known as the key distribution problem: both parties must share a secret key before communicating securely. As networks grew, managing these keys securely became increasingly difficult and error-prone.

The conceptual breakthrough came in 1976 with the Diffie–Hellman key exchange, which introduced the notion of public-key cryptography. Their system enabled two parties to establish a shared secret over an insecure channel. However, it did not provide encryption or signatures. RSA filled this gap by providing both encryption and digital signing within a unified mathematical framework.

RSA rapidly became the cornerstone of Public-Key Infrastructure (PKI), supporting HTTPS, SSL/TLS certificates, secure email, software signing, and authentication systems. Despite newer algorithms emerging, RSA remains widely deployed due to its long-standing trust, ease of implementation, and compatibility with existing systems.

3 Mathematical Foundations of RSA

RSA begins with the generation of two large prime numbers, p and q, selected randomly and independently. Their product,

$$n = pq,$$

forms the RSA modulus. The security of RSA relies on the computational difficulty of factoring this large semiprime.

Euler's totient function is then computed:

$$\phi(n) = (p-1)(q-1).$$

A public exponent e is chosen such that $\gcd(e, \phi(n)) = 1$. The commonly used value $e = 65537$ provides efficiency and security. The private exponent d is computed as the modular inverse of $e \mod \phi(n)$:

$$ed \equiv 1 \pmod{\phi(n)}.$$

Encryption transforms a message m into ciphertext:

$$c = m^e \bmod n.$$

Decryption recovers the plaintext:

$$m = c^d \bmod n.$$

RSA's correctness is guaranteed by Euler's theorem.

4 RSA Digital Signatures

Digital signatures represent one of the most important uses of RSA and are arguably even more influential today than RSA encryption itself. While RSA encryption provides confidentiality by allowing anyone to encrypt a message for the private key holder, RSA signatures provide additional properties that are essential for security. These include authenticity, which ensures that the message truly originates from the claimed sender; integrity, which ensures that the message has not been altered; and non-repudiation, which prevents the signer from denying authorship. RSA signatures rely on the same mathematical operations as RSA encryption but apply them in the reverse order for verification.

4.1 Signing Process

In a standard RSA signature scheme, the signer begins with a message m, which is first processed through a cryptographic hash function such as SHA-256. Hashing produces a fixed-length digest h that represents the message. The signer uses their private exponent d to compute:

$$s = h^d \bmod n.$$

Only someone who possesses the private key can generate this value. Proper padding, such as RSA-PSS, is required to prevent signature forgery attacks.

4.2 Verification Process

To verify a signature, the recipient first hashes the received message to obtain its digest h, Using the public key (e, n), the verifier computes the following:

$$h' = s^e \bmod n.$$

If h' matches the actual digest of the message (h), the signature is valid. This process allows anyone with the public key to confirm the authenticity of the message, demonstrating the power of RSA signatures in distributed environments.

4.3 Use in Certificates and PKI

Digital signatures are fundamental to the operation of public key infrastructure (PKI). Certificate authorities (CAs) rely on RSA signatures to certify the authenticity of public keys used by websites and organizations. Web browsers contain a list of trusted CA public keys, allowing them to verify website certificates and ensure secure HTTPS connections. Many systems that require proof of identity or integrity, such as software updates, secure email, document signing, and operating system verification, depend on RSA signatures.

4.4 Attack Implications

An improper implementation of RSA signatures can lead to security failures. Weak randomness, insecure padding schemes, or outdated hashing algorithms can allow attackers to forge signatures or manipulate message contents. Historical attacks, including Bleichenbacher-style signature forgery and hash collision exploitation, demonstrate that secure digital signatures depend on both strong cryptography and disciplined engineering practices.

5 Main Issues, Risks, and Real-World Problems

Although RSA is mathematically sound, its practical security depends on careful parameter selection, robust randomness, and proper implementation. One significant concern is the size of the key. Keys shorter than 1024 bits are vulnerable to factorization attacks using modern hardware. Even 1024-bit keys are considered weak for long-term security, and many standards now require minimum sizes of 2048 bits or higher.

Another concern involves randomness. RSA key generation requires the selection of unpredictable primes. If an implementation uses a poor entropy source, multiple devices may generate keys that share a prime factor. In such cases, the modulus n can be factored immediately using a simple check for the greatest common divisor. Studies of real-world RSA deployments have identified thousands of such weak keys.

Padding is another major source of vulnerability. Without secure padding, RSA encryption and signatures become deterministic and susceptible to chosen-ciphertext or chosen-message attacks. Historical padding techniques, particularly those used before the adoption of RSA-OAEP and RSA-PSS, exposed systems to dangerous attack vectors.

Side-channel attacks present an additional threat. These attacks exploit information leaked by physical devices during computation, such as timing variations, fluctuations in power consumption, or electromagnetic emissions. If an attacker can observe these patterns, they may be able to recover RSA private keys.

Finally, advances in quantum computing present a long-term threat to RSA. The Shor algorithm provides an efficient method to factor in large semiprimes, which would make RSA insecure once sufficiently powerful quantum hardware becomes available.

6 RSA in TLS and Hybrid Cryptographic Systems

RSA has played a central role in securing communication through the Transport Layer Security (TLS) protocol. For many years, RSA was used both for authentication and for the exchange of symmetric session keys. Its widespread adoption was due to simplicity, interoperability, and the availability of hardware and software that supported RSA operations.

6.1 RSA in the TLS Handshake

In earlier versions of TLS, particularly TLS 1.0 through TLS 1.2, one of the most common handshake methods relied on RSA key exchange. In this scheme, the client retrieves the server's certificate, verifies the signature on the certificate, and extracts the server's RSA

public key. The client then generates a random 48-byte pre-master secret, encrypts it using the public key of the server, and sends the result to the server. Because only the server possesses the private key, it is the only party able to decrypt the pre-master secret. Then both sides derive symmetric session keys for encrypting the remainder of the communication.

6.2 Why RSA Key Exchange Is Deprecated

Although RSA key exchange was widely deployed, it has several weaknesses. The most important limitation is its lack of forward secrecy. If a server's RSA private key is compromised, an attacker who has recorded past encrypted sessions can retroactively decrypt them. This creates long-term confidentiality risks, especially for systems that maintain logs or operate in sensitive environments.

RSA key exchange also exposes servers to side-channel attacks because every handshake involves performing a private-key decryption operation. In contrast, the Diffie-Hellman elliptic curve key exchange, now mandated in TLS 1.3, allows servers to authenticate themselves with RSA signatures while avoiding RSA decryption operations during key establishment.

6.3 Hybrid Protocols

Modern TLS deployments use a hybrid model in which RSA remains important for authentication through digital signatures, while key agreement and bulk encryption rely on symmetric or Diffie–Hellman–based algorithms. In this model, RSA establishes identity, symmetric cryptography provides performance, and ephemeral Diffie–Hellman ensures forward secrecy. This division of responsibilities results in faster, safer, and more resilient communication than traditional RSA-only handshakes.

7 Real RSA Security Incidents

A number of significant security incidents have demonstrated that RSA can fail dramatically when improperly implemented. These incidents reveal how sensitive RSA deployments are to mistakes involving randomness, padding, or key generation hardware.

7.1 Debian OpenSSL Vulnerability (2008)

In 2008, a critical flaw was discovered in the Debian Linux implementation of OpenSSL. A small code modification inadvertently removed most of the entropy from the random number generator used to create keys. As a result, millions of RSA keys were generated from a tiny

set of possible values. Attackers were able to precompute all of these keys and compromise affected systems instantly. This incident highlighted the importance of secure randomness in cryptographic key generation.

7.2 ROCA Attack (2017)

The ROCA vulnerability affected RSA keys generated by Infineon Trusted Platform Modules. A deterministic algorithm was used to generate primes, and this algorithm introduced a mathematical pattern that reduced the effective security of the resulting RSA moduli. Researchers demonstrated that these moduli could be factored much more efficiently than normally expected. Millions of passports, identity cards, laptops, and security tokens were affected, and widespread key replacement efforts were required.

7.3 IoT Weak-Key Study

A large-scale study by Heninger and colleagues investigated real RSA keys deployed across the internet. The researchers found that approximately 0.4 percent of RSA keys shared a prime factor with another key, indicating a failure in entropy generation. These weak keys allowed immediate factorization of the modulus using greatest common divisor computations. Many of the affected systems were embedded or poorly maintained IoT devices.

8 Demonstration of Weak RSA

To illustrate how fragile RSA becomes when key parameters are insufficient, a demonstration was performed using intentionally small primes. Two 16-bit primes were selected, and a corresponding RSA modulus was generated. Although the algorithm was implemented correctly, the extremely small key size produced a modulus that could be factored in a negligible amount of time.

A ciphertext was then produced using the public key. An attacker who knew only the public key and the ciphertext attempted to factor the modulus using Pollard's Rho algorithm. The modulus was factored almost instantly. Once the primes were recovered, the attacker recomputed the totient, derived the private exponent, and successfully decrypted the ciphertext. This experiment demonstrates that RSA does not degrade gradually when parameters are weak. Instead, it collapses entirely once the primes become small or predictable.

9 Discussion

The analysis of RSA shows that while the algorithm is mathematically robust, its real-world security depends on strong implementation practices. Issues such as weak randomness, insufficient key lengths, and outdated padding schemes can lead to catastrophic failures. Side-channel vulnerabilities further complicate secure deployment and require specialized countermeasures such as constant-time algorithms, masking techniques, and hardware shielding.

The weak RSA demonstration confirms the critical role of parameter selection. Even when RSA is implemented correctly, small primes or predictable random number generation render the system insecure. Modern deployments must therefore follow strict guidelines for key generation, randomness, padding, and protocol design. The overall security of RSA depends on the integrity of the entire system surrounding it, not merely its mathematical foundation.

10 Future of RSA and the Post-Quantum Transition

The long-term viability of RSA is challenged by advancements in quantum computing. Shor's algorithm provides a polynomial-time method for factoring large semiprimes, meaning that RSA would become ineffective once sufficiently powerful quantum computers are available. Although practical quantum computers capable of breaking RSA do not yet exist, significant research investment suggests that this capability may eventually emerge.

Organizations are preparing for this transition by exploring post-quantum cryptography (PQC). The National Institute of Standards and Technology (NIST) has selected several algorithms for standardization, including CRYSTALS–Kyber for encryption and CRYSTALS–Dilithium for digital signatures. These algorithms are designed to resist attacks from both classical and quantum adversaries.

A gradual transition is expected. Hybrid certificates that combine RSA with PQC algorithms are being introduced to maintain compatibility with existing systems while providing protection against future quantum threats. Although RSA will continue to exist in legacy systems for many years, new deployments are increasingly shifting toward cryptographic primitives that can provide long-term post-quantum security.

11 Conclusion

RSA remains one of the most influential cryptographic systems ever developed, and its impact on the evolution of secure digital communication cannot be overstated. For more than four decades, RSA has enabled confidential messaging, authenticated online transactions, and the development of large-scale public key infrastructures that support modern internet security. Its elegant mathematical construction, rooted in number theory and modular exponentiation, provides a mechanism through which trust can be established even when communicating parties have no prior relationship. This property fundamentally reshaped how secure communication is implemented across global networks.

The analysis presented throughout this report demonstrates both the strengths and vulnerabilities of RSA. On the positive side, RSA provides strong theoretical security when implemented with sufficiently large key sizes and modern padding schemes. Its mathematical foundations remain difficult to attack through classical computational methods, and its flexibility allows it to support both encryption and digital signatures. These features have made RSA a cornerstone of protocols such as TLS, secure email, and digital certificates.

However, the practical security of RSA is not determined solely by its mathematical structure. Real-world deployments reveal that its safety heavily depends on factors such as the quality of randomness during key generation, the proper use of secure padding methods, the avoidance of deterministic or structurally predictable behavior, and resilience against side-channel attacks. Incidents such as the Debian OpenSSL vulnerability, the ROCA attack, and widespread weak-key findings in IoT devices show that RSA can fail completely when even one part of the implementation pipeline is compromised. The weak RSA demonstration in this report reinforces this observation by illustrating how quickly a poorly configured system collapses under attack.

Furthermore, RSA faces ongoing challenges from protocol design and long-term cryptographic planning. The deprecation of RSA key exchange in TLS reflects the broader industry shift toward systems that offer forward secrecy by default. While RSA continues to play an important role in authentication, newer protocols increasingly rely on hybrid approaches that separate identity verification from secure key establishment. This trend will likely accelerate as cryptographic systems evolve to meet emerging security requirements.

Looking ahead, the most significant threat to RSA arises from quantum computing. Shor's algorithm presents a theoretical ability to break RSA by efficiently factoring large semiprimes, which would undermine the security of all RSA-based systems. Although large-scale quantum computers capable of executing such attacks do not yet exist, ongoing research and investment indicate that cryptographers must prepare for this possibility. The transition

to post-quantum cryptography is therefore not simply an academic exercise but an essential step toward ensuring the long-term security of digital communication infrastructures.

In conclusion, RSA remains an essential component of modern cryptographic practice, but it also serves as a cautionary example of the complexity of real-world security. Strong cryptography requires not only sound mathematics but also rigorous implementation, high-quality randomness, resistant protocols, and careful attention to emerging computational threats. As the field continues to evolve, the lessons learned from studying RSA will remain relevant for understanding both the potential and the limitations of cryptographic systems. Although RSA may eventually be replaced in certain contexts by post-quantum algorithms, its conceptual and historical significance ensures that it will continue to influence the design and evaluation of secure communication technologies for many years to come.

Please visit our GitHub repository: $https://github.com/Taylor-Hunter/RSA_Project$ for demos we created.

References

[1] Almazari, Mahmoud M., et al. "RSA Private Keys and the Presence of Weak Keys: An Evaluation." *Journal of Discrete Mathematical Sciences and Cryptography*, July 2022.

[2] Boneh, Dan. "Twenty Years of Attacks on the RSA Cryptosystem." *Notices of the AMS*, 2007.

[3] Gerjuoy, Edward. "Shor's Factoring Algorithm and Modern Cryptography." *American Journal of Physics*, vol. 72, no. 5, 2004.

[4] Heninger, Nadia, et al. "Detection of Widespread Weak Keys in Network Devices." *USENIX Security Symposium*, 2012.

[5] Just, Jiri, and John Coffey. "An Assessment of Attacks Strategies on the RSA Public-Key Cryptosystem." *International Institute of Informatics and Systemics*, 2009.

[6] Kilgallin, Jonathan, and Ross Vasko. "Factoring RSA Keys in the IoT Era." IEEE / Keyfactor, 2019.

[7] Maitra, Subhamoy, and S. Sarkar. "Revisiting Wiener's Attack: New Weak Keys in RSA." *Lecture Notes in Computer Science*, vol. 5222, Springer, 2008.

[8] Paar, Christof, Jan Pelzl, and Tim Güneysu. *Understanding Cryptography: From Established Symmetric and Public-Key Primitives to Advanced Protocols*. 2nd ed., Springer Spektrum, 2024.

[9] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM*, vol. 21, no. 2, 1978.

[10] Ruzai, W. N. A., et al. "Concurrent Factorization of RSA Moduli via Weak Key Conditions." *AIMS Mathematics*, vol. 9, no. 5, 2024.