

THE 21ST EDITION OF THE INTERNATIONAL CONFERENCE
EUROPEAN INTEGRATION
REALITIES AND PERSPECTIVES**Comparative Analysis of the Merkle–Hellman Knapsack and RSA
Public-Key Cryptosystems: Efficiency vs. Security****Ciprian Rusescu¹**

Abstract: This paper provides a targeted comparative analysis of the Merkle–Hellman knapsack cryptosystem and the RSA public-key scheme, aiming to evaluate their relative efficiency and practical relevance. The research investigates key generation complexity, encryption and decryption performance, and resistance to established cryptanalytic techniques. Based on an experimental implementation and benchmarking across different key sizes, the result indicates that the Merkle–Hellman approach achieves lower computational overhead in both encryption/decryption process, due to its reliance on super-increasing sequences. However, these efficiency advantages are counterbalanced by structural weaknesses in the key transformation process, which expose the system to polynomial-time attacks. By comparison, RSA, despite higher computational costs, preserves robust security grounded in the hardness of integer factorization. Overall, the analysis demonstrates the conclusion that Merkle–Hellman, despite its historical importance, fails to achieve a sustainable efficiency-security balance for modern applications, reinforcing RSA’s dominance in practical public-key cryptography.

Keywords: Public-key cryptography; RSA; Merkle – Hellman cryptosystem; Subset sum problem; Algorithmic complexity

JEL Classification: C02; C63; D83; O33

1. Introduction

Public-key cryptography represents one of the most significant breakthroughs in the information security history, enabling secure communication over open networks without the necessity of a sharing the secret key. Extensively studied since the inception (dated on the late 1970s), asymmetric cryptography has become a cornerstone of modern digital infrastructures, underpinning secure web communication, authentication protocols, digital signatures, and also public key infrastructures (PKI). The security level of such systems critically depends on the careful selection of underlying mathematical problems, computationally efficient for legitimate users but also infeasible to solve for adversaries.

Early research in public-key cryptography explored a wide range of mathematical foundations, including number theory, combinatorics, and complexity theory. Knapsack-based cryptosystems were among the first proposed schemes, whilst notably the Merkle–Hellman knapsack cryptosystem sought to exploit

¹ Senior Lecturer, Danubius International University, Galati, Romania, Address: Bd. Galati, nr. 3, Galati, Corresponding author: ciprian.rusescu@univ-danubius.ro.



Copyright: © 2026 by the authors.
Open access publication under the terms and conditions of the
Creative Commons Attribution-NonCommercial (CC BY NC) license
(<https://creativecommons.org/licenses/by-nc/4.0/>)

the presumed hardness of the subset-sum problem (Merkle & Hellman, 1978). In its general form, the subset-sum problem is NP-complete, being an attractive option for cryptographic applications. The Merkle–Hellman system aimed to value this theoretical hardness while maintaining efficient encryption/decryption by using superincreasing sequences.

Proposed in 1978, the Merkle–Hellman algorithm represents one of the first practical implementations of asymmetric cryptography, triggering a wave of research into alternative combinatorial-based schemes. Early studies were focused on exploring various alternatives of the superincreasing knapsack problem and also methods who target its apparent complexity improvement, through modular transformations/permutations of the private key sequence (Lagarias, 1983; Odlyzko, 1990). Although these tentatives initially enhanced the system perceived security, they ultimately failed to withstand rigorous cryptanalysis.

The Merkle–Hellman cryptosystem appeal consists in its conceptual simplicity and computational efficiency. By embedding a superincreasing private sequence into an apparently random public key through modular multiplication and permutation, the system was projected to veil its trapdoor structure while enabling fast decryption for the legitimate key holder. At that time, current approach was considered a promising alternative to number-theoretic schemes, especially in environments with limited computational resources.

However, subsequent cryptanalysis revealed a fundamental mismatch between theoretical problem hardness and presumed practical cryptographic security. Although the general subset-sum problem is a NP-complete one, the specific patterns generated by the Merkle–Hellman construction highlight strong structural properties that significantly reduce their effective complexity. Shamir proved that these structural weaknesses could be efficiently exploited to recover the private key in polynomial time, effectively kneeling the original cryptosystem (Shamir, 1984). This result underscored an essential lesson in cryptography, namely that security depends not only on the underlying problem difficulty, but also on the absence of exploitable structure in the public representation. Subsequent research confirmed that even extended variants of Merkle–Hellman were vulnerable unless substantial modifications were introduced, and these modifications often compromised computational efficiency (Brickell, 1984; Stern, 1993).

In contrast to knapsack-based structures, the RSA cryptosystem relies on the computational difficulty of factoring large composite integers, a problem that has resisted polynomial-time solutions for classical computers despite decades of intensive study (Rivest et al., 1978). Introduced by Rivest, Shamir and Adleman in 1978, the system became very quickly the standard for public – key cryptography, because security assumptions were very well understood. But the major advantage obviously consisted in the fact that its design has benefited from intense cryptanalytic scrutiny. Numerous studies have explored RSA’s performance, key generation algorithms, and optimizations for modular exponentiation (Boneh, 1999; Shoup, 2009).

2. Literature Review

Comparative works have examined the divergent trajectories of Merkle–Hellman and RSA, highlighting the central trade-off in public key cryptography: efficiency versus robustness (Menezes, van Oorschot & Vanstone, 1996). More precisely, differences between knapsack-based schemes and RSA were analyzed, emphasizing efficiency advantages of knapsack constructions while reaffirming the superior security of number-theoretic approaches under classical computing assumptions (Schneier, 1996).

Modern surveys also contextualize these findings within the broader evolution of asymmetric cryptography, noting that lessons learned from Merkle–Hellman influenced later developments in lattice-based and post-quantum cryptography (Micciancio & Regev, 2009).

Merkle–Hellman cryptosystem requires lower computational effort due to its reliance on simple arithmetic operations, but its efficiency is achieved with the price of introducing structural regularities that finally undermine security. In return, RSA accepts higher computational complexity in exchange for a security foundation less susceptible to structural attacks. In other words, although RSA is computationally more demanding than early knapsack-based schemes, its resilience/predictable security properties have contributed to its widespread adoption and long-term viability in real-world applications.

Taken together, the foundational works on the Merkle–Hellman knapsack cryptosystem and RSA laid the groundwork for public key cryptography, while simultaneously illustrating two different security trajectories. Early research on knapsack-based schemes quickly revealed significant structural weaknesses: in particular, the original trapdoor knapsack construction proposed by Merkle and Hellman was shown to be vulnerable to attacks exploiting the low-density structure of the underlying subset sum problem. Lattice-based techniques developed by Lagarias and Odlyzko demonstrated that, despite the NP-hardness of the general subset sum problem, specific structured instances such as Merkle–Hellman fail to provide practical cryptographic security (Lagarias & Odlyzko, 1985; Joux & Stern, 1991). As a result, knapsack cryptosystems gradually lost prominence as viable public key primitives, remaining primarily of historical and pedagogical interest.

In contrast, the RSA cryptosystem has withstood decades of cryptanalytic scrutiny and remains a central reference point in public key cryptography. Introduced by Rivest, Shamir, and Adleman, RSA derives its security from the presumed difficulty of factoring large composite integers, a problem that has resisted efficient general solutions in the classical computing model (Rivest, Shamir & Adleman, 1978). Subsequent analysis established that while RSA is theoretically sound, its practical security depends critically on correct parameter selection and key generation procedures. Classical treatments emphasize that vulnerabilities typically arise not from the underlying hardness assumption, but from improper implementation choices, such as poorly chosen primes or insufficient randomness during key generation (Stinson, 2006).

Beyond their immediate practical relevance, the contrasting histories of Merkle–Hellman and RSA highlight important lessons in public key design. Whereas Merkle–Hellman was compromised by exploitable algebraic structure, RSA has largely resisted attacks except under restrictive assumptions, such as unusually small private exponents. Classical cryptanalytic results, including Wiener’s attack on short secret exponents and later refinements using lattice techniques, helped delineate the conditions under which RSA can be considered secure (Wiener, 1990; Boneh & Durfee, 2000). These comparative observations clarify why RSA emerged as a long-term standard, while knapsack-based constructions were largely abandoned. While this discussion focuses on classical constructions, the lessons drawn from Merkle–Hellman and RSA continue to inform modern cryptographic standards and evaluations (Katz & Lindell 2015, 2020; NIST 2024).

Against this historical and conceptual background, the present paper targets a comparative evaluation of the Merkle–Hellman knapsack/RSA cryptosystems, examining both performance features and security implications. Through an analysis of key generation process, encryption/decryption efficiency and resistance to cryptanalytic techniques known so far, the study aims to reassess the practical applicability of Merkle–Hellman in the light of modern security needs. Beyond a purely historical

comparison, this work underlines the broader design lessons derived from early public-key failures and discusses their relevance to contemporary cryptographic practice.

By reconsidering the Merkle–Hellman cryptosystem as a case study in cryptographic vulnerability, the paper tries to highlight the importance of structural soundness in public-key design and to explain why RSA continues to represent a viable and widely deployed solution in practical cryptography.

3. From Theory to Practice

Previous discussion has sketched the comparative performance and security properties of the Merkle–Hellman and RSA cryptosystems but a better understanding of these aspects requires internal mechanisms examination through concrete examples. First one simply illustrates the Merkle–Hellman knapsack construction, reflecting the step-by-step process in an accessible manner, pointing out the main idea of the Shamir’s attack. Next, our focus is moving to an analogous example for RSA, highlighting the relevance of Wiener’s attack. These gathered demonstrations connect theoretical principles to practical outcomes, showing how structural design choices translate into cryptographic vulnerabilities.

Let’s consider for example the superincreasing sequence

$$a = (3, 5, 12, 25, 53)$$

containing 5 elements, also the positive integer $q = 101$, greater than the sum of all components. ($q > \sum_{i=1}^5 a_i$). We fix the natural number $r = 29$ as a prime number related to 101, $\gcd(29, 101) = 1$. By using the sequence $b_i = r \cdot a_i, i = \overline{1, n}$, we calculate $b = (87, 44, 45, 18, 22)$ and decide, for example, to encrypt the binary message $m = (1, 0, 1, 1, 0)$. The result will be

$$C = 1 \cdot 87 + 0 \cdot 44 + 1 \cdot 45 + 1 \cdot 18 + 0 \cdot 22 = 150.$$

As we previously mentioned, the encryption process involves simple mathematic transformations, whose simplicity trying to be masked by some modular transformations. Same rationality is hinted in the decryption process, making use to the r inverse value, considering modular q operations. ($u = r^{-1} \bmod q$).

Approached from a cryptanalytic perspective, Shamir’s attack exploits the arithmetic structure of the Merkle–Hellman public key, using rational approximation techniques, such as continued fractions. Starting from public key elements and the modulus, the attack considers a_i/n ratios form, targeting the approximation of the hidden multiplier divided by the modulus.

The continued fraction expansion of these ratios produces a sequence of convergents that reveal particularly accurate approximations with small denominators. In many instances, these convergents/semiconvergents yield feasible candidates for the secret parameters, able to define the private super increasing sequence. Exhaustive search over all subset sums is avoided this way, relying on the well-known optimality properties of continued fraction approximations.

Good to emphasize that continued fractions do not introduce additional assumptions, they rather make explicit the mathematical mechanism by which the public key leaks information about an equivalent private one. Thus, the attack efficiency springs from the Merkle–Hellman scheme inherent structure, allowing efficient recovery of private key data from the public key itself.

The bellow presented diagram illustrates the decryption process once the modular parameters r and q have been recovered — typically via continued fractions — reducing the public key to an equivalent superincreasing sequence and enabling efficient message recovery.

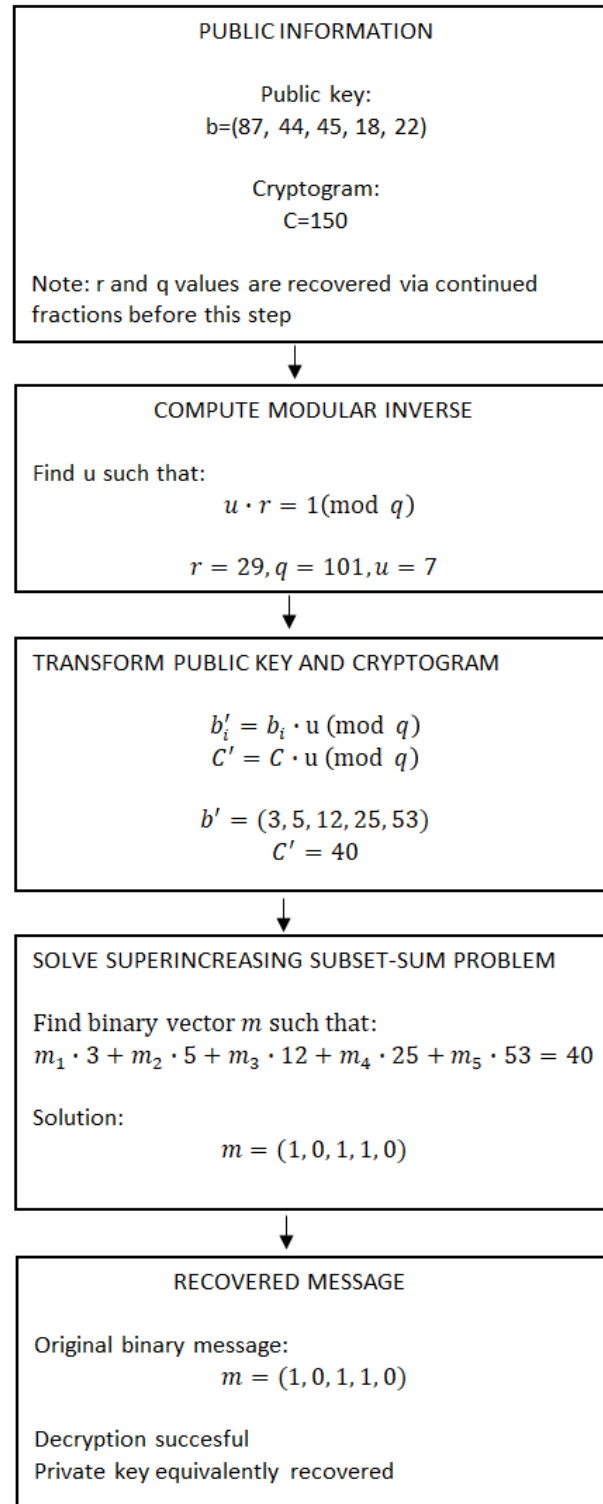


Figure 1. Shamir attack on the Merkle–Hellman cryptosystem

After seeing how Merkle–Hellman works through the previous numerical example, we now turn our attention to RSA to show another practical approach. Like Merkle–Hellman, RSA allows secure message encryption, but it uses a different underlying method. In the following instance, we will outline each step of encryption/decryption using real numbers. This will illustrate not only how the algorithm transforms the message but also the role of the keys in ensuring security. By following the process step by step, the mechanics of RSA become clear, preparing us to compare it with the earlier Merkle–Hellman demonstration.

The strength of RSA's security relies on integer factorization. Using a small numeric example, we demonstrate its practical encryption/decryption mechanics. Two natural primes are randomly chosen first, $p = 53, q = 59$, resulting immediately

$$n = p \cdot q = 3127$$

and afterwards Euler totient function

$$\phi(n) = (p - 1) \cdot (q - 1) = 3016$$

Giving the necessity that public exponent to be chosen from the $\phi(n)$ nondivisor set, will be considered $b = 17$ value as public exponent. By using Euclid algorithm extension, the private exponent $a = 2129$ will be calculate, and the encryption becomes:

$$1234^{17} \bmod 3127 = 2501$$

where $m = 1234$ represent the message to be sent.

The decryption reverse process works in the same way:

$$2501^{2129} \bmod 3127 = 1234$$

RSA encryption/decryption process involves modular exponentiation, performed by using the square-and-multiply algorithm, which efficiently calculates large powers modulo n by leveraging the binary representation of the exponent. Each step treats one exponent bit, starting from the most significant to the least significant (the original message m encryption/ciphertext c decryption). This method prevents working directly with unwieldy large numbers, clearly demonstrating how the plaintext message transforms into ciphertext and how the ciphertext is reverted to the original message.

The tables below clearly illustrate the complete RSA modular exponentiation, for both encryption and decryption processes. Each intermediate result remains modulo n , ensuring manageable values throughout. This stepwise algorithm highlights the efficiency and correctness of RSA, even when dealing with large exponents and moduli.

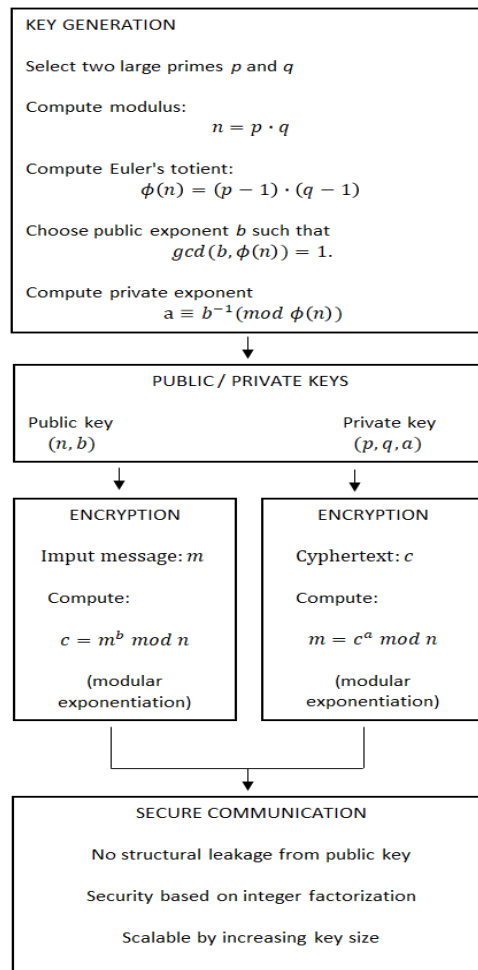
Table 1. RSA Encryption ($b = 17 = 10001_2, m = 1234, n = 3127$)

Step i	Bit c_i	Square and multiply	$m^b \pmod n$
4	1	$1^2 \cdot 1234 = 1234$	1234
3	0	$1234^2 = 3034$	3034
2	0	$3034^2 = 2395$	2395
1	0	$2395^2 = 1107$	1107
0	1	$1107^2 \cdot 1234 = 2501$	2501

Table 2. RSA Decryption ($a = 2129 = 100001010001_2$, $c = 2501$, $n = 3127$)

Step i	Bit c_i	Square and multiply	$m^b \pmod n$
11	1	$1^2 \cdot 2501 = 2501$	2501
10	0	$2501^2 = 1001$	1001
9	0	$1001^2 = 1361$	1361
8	0	$1361^2 = 1137$	1137
7	0	$1137^2 = 1318$	1318
6	1	$1318^2 \cdot 2501 = 2769$	2769
5	0	$2769^2 = 3084$	3084
4	1	$3084^2 \cdot 2501 = 2643$	2643
3	0	$2643^2 = 2858$	2858
2	0	$2858^2 = 440$	440
1	0	$440^2 = 2853$	2853
0	1	$2853^2 \cdot 2501 = 1234$	1234

A more detailed scheme of the key process generation, public/private key separation, and encryption/decryption via modular exponentiation is presented below. Unlike knapsack-based schemes, RSA does not expose exploitable structural information through its public key.

**Figure 2. Conceptual flow of the RSA cryptosystem**

In contrast to the Merkle–Hellman scheme, whose linear structure introduces exploitable regularities, RSA does not expose comparable structural leakage under standard parameter choices. The public key

(N, b) does not reveal direct linear relations linking its components to the private exponent a , whose recovery from public information is computationally infeasible.

Wiener's cryptanalytic technique works only when the private exponent a is unusually small. The attack relies on the continued fraction expansion of the rational number b/n , whose convergents reveal rational approximations k/a satisfying the relation $b \cdot a - k \cdot \varphi(n) = 1$. When $3a < n^{\frac{1}{4}}$ these convergents may directly expose the private exponent.

Outside this "dangerous" parameter range, the convergents/semiconvergents of the continued fraction for b/n no longer correspond to valid solutions, the method becoming ineffective. In the present scenario, continued fractions do not exploit a structural weakness of RSA itself, but rather the presence of atypical key parameters. As long as RSA keys are generated according to standard guidelines (sufficiently large primes and a private exponent of adequate size) continued fraction attacks, including Wiener's method, will not compromise its security.

As a general rephrasing of all the above, while Merkle–Hellman's algebraic structure inherently facilitates cryptanalysis, RSA remains secure under classical assumptions as long as its parameters are chosen correctly, underscoring the critical role of structural design and parameter selection in public-key cryptography.

4. Conclusions

The Merkle–Hellman knapsack cryptosystem stands out historically as one of the first practical public-key schemes, valued for simplicity/efficiency. Superincreasing sequences use allows rapid decryption, while the underlying concepts are relatively easy to understand and implement. This combination of accessibility and computational speed made Merkle–Hellman a key educational example, revealing the principles of public-key cryptography and inspiring further field research.

However, the system's elegance cannot hide significant vulnerabilities. Structural weaknesses were revealed by Shamir's attack, highlighting that the public key contains enough information to reconstruct a private key in polynomial time. Attempts to enhance security through larger keys proved to be insufficient, as its own algebraic structure remained exploitable. Accordingly, Merkle–Hellman gradually lost practical relevance, surviving so far mainly as a teaching tool rather than a basis for secure communication.

In return, RSA offers a more robust security foundation, grounded in the difficulty of factoring large integers. Its resilience under many decades of cryptanalytic scrutiny, combined with widespread adoption in applications such as SSL/TLS, digital signatures, and public key infrastructures, underscores its lasting utility. Efficient algorithms for modular exponentiation, alongside the ability to adjust key sizes according to security requirements, have allowed RSA to balance theoretical strength with practical usability, consolidating its role as a cornerstone of modern cryptography.

Despite all these rigorous arguments, RSA is not exempt from trade-offs. Key generation as well as encryption operations are computationally heavier than those of Merkle–Hellman, whilst larger keys demand greater memory/bandwidth. Moreover, emerging quantum computing techniques, mainly Shor's algorithm, represents potential future threats to its security assumptions. Though, careful parameter choice and standardized implementations have allowed RSA to keep its practical security, proving that a cryptosystem success depends not only on the hard mathematical problems but also the resilient design and meticulous real-world deployment.

Based on this comparison, the contrasting trajectories of Merkle–Hellman and RSA further highlight the decisive role of structural robustness in public-key cryptography. Although Merkle–Hellman offers fast decryption and a high level of conceptual simplicity, its reliance on superincreasing sequences induce exploitable patterns that ultimately undermine security. While more computationally demanding, RSA avoids such structural problems, relying on the well-studied hardness of integer factorization. This difference explains why Merkle–Hellman remains largely of historical/educational interest, whilst RSA has achieved standardization and widespread practical adoption.

From a broader perspective, this contrast reflects a recurring trade-off between efficiency and security. Systems optimized primarily for computational speed may expose specific algebraic structures that attackers can exploit (see Shamir’s attack). In return, cryptosystems such as RSA show that higher computational costs acceptance can lead to much stronger/more durable security guarantees, particularly when scalability is achieved through adjustable key sizes rather than fragile structural assumptions.

These remarks also offer important lessons for modern and emerging cryptographic design. Theoretical problem hardness by its own is not enough if public keys leak meaningful patterns, this insight being especially relevant for contemporary research into post-quantum and lattice-based schemes. Future cryptosystems must therefore balance efficiency with rigorous structural resilience, ensuring that performance gains do not harm of long-term security.

Finally, the comparison reinforces the enduring educational value of well-chosen numerical examples. Even when a cryptosystem is no longer secure in practice, simplified instances remain powerful tools for illustrating cryptographic mechanisms and design principles. Extending such examples in simulations or experimental frameworks may further enhance understanding, bridging the gap between abstract theory and real-world cryptographic deployment.

References

- Boneh, D. (1999). Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2), 203–213.
- Boneh, D., & Durfee, G. (2000). Cryptanalysis of RSA with private key d less than [missing value]. *IEEE Transactions on Information Theory*, 46(4), 1339–1349.
- Brickell, E. F. (1984). Solving low density knapsacks. In *Advances in cryptology, Proceedings, CRYPTO 83* (pp. 25–37). Plenum Press.
- Joux, A., & Stern, J. (1991). Improving the critical density of the Lagarias–Odlyzko attack against subset sum problems. *Lecture Notes in Computer Science*, 529, 258–264.
- Katz, J., & Lindell, Y. (2015). *Introduction to modern cryptography* (2nd ed.). Chapman & Hall/CRC.
- Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography*. CRC Press.
- Lagarias, J. C. (1983). The knapsack problem and public-key cryptography. In *Advances in cryptology: Proceedings of CRYPTO 82* (Lecture Notes in Computer Science, Vol. 196, pp. 3–21). Springer.
- Lagarias, J. C., & Odlyzko, A. M. (1985). Solving low-density subset sum problems. *Journal of the Association for Computing Machinery*, 32(1), 229–246.
- Menezes, A., van Oorschot, P., & Vanstone, S. (1996). *Handbook of applied cryptography*. CRC Press.
- Merkle, R., & Hellman, M. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5), 525–530.
- Micciancio, D., & Regev, O. (2009). Lattice-based cryptography. In *Post-quantum cryptography* (pp. 147–191). Springer.

National Institute of Standards and Technology. (2024). *Digital signature standard (DSS): FIPS PUB 186-5*. National Institute of Standards and Technology.

Odlyzko, A. M. (1990). The rise and fall of knapsack cryptosystems. In *Cryptology and computational number theory* (Proceedings of Symposia in Applied Mathematics, Vol. 42, pp. 75–88). American Mathematical Society.

Rivest, R., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120–126.

Schneier, B. (1996). *Applied cryptography* (2nd ed.). Wiley.

Shamir, A. (1984). A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, IT-30(5).

Shoup, V. (2009). *A computational introduction to number theory and algebra* (2nd ed.). Cambridge University Press.

Stern, J. (1993). A new identification scheme based on syndrome decoding. In *Advances in cryptology—CRYPTO '93* (Lecture Notes in Computer Science, Vol. 773, pp. 13–21). Springer.

Stinson, D. R. (2006). *Cryptography: Theory and practice* (3rd ed.). Chapman & Hall/CRC.

Wiener, M. J. (1990). Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3), 553–558.