Padding Schemes for RSA and their Security

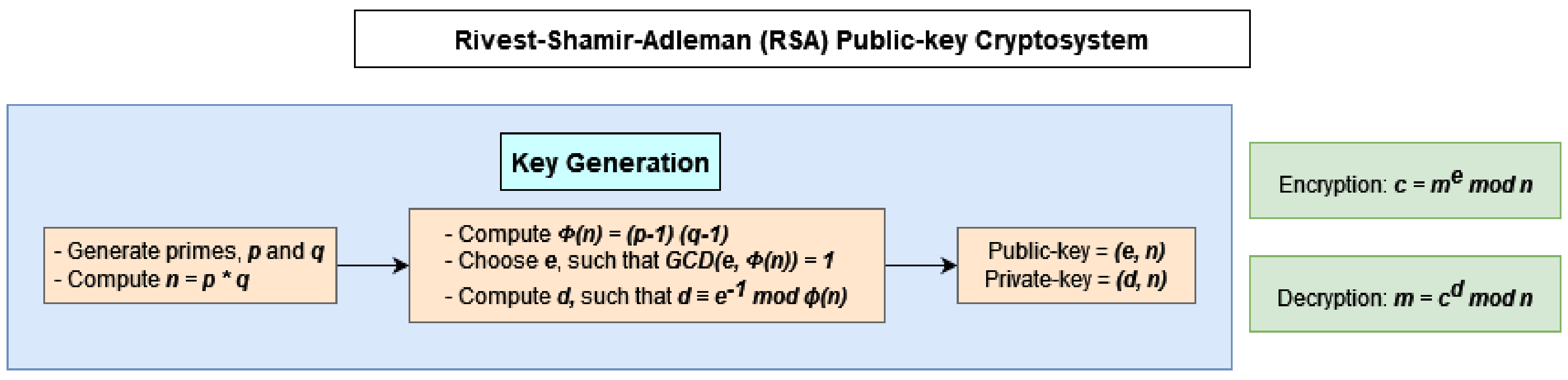
Calvin Ryan Dsouza (cd5184@rit.edu)

Advisor: Stanisław P. Radziszowski

Computer Science @ RIT
B.Thomas Golisano College of Computing & Information Sciences

INTRODUCTION

- One of the oldest algorithms used for data transmission and digital signatures is the Rivest-Shamir-Adleman (RSA) public-key cryptosystem. The security of RSA banks on difficulty of factoring the product of 2 large prime numbers i.e., the factorization problem.

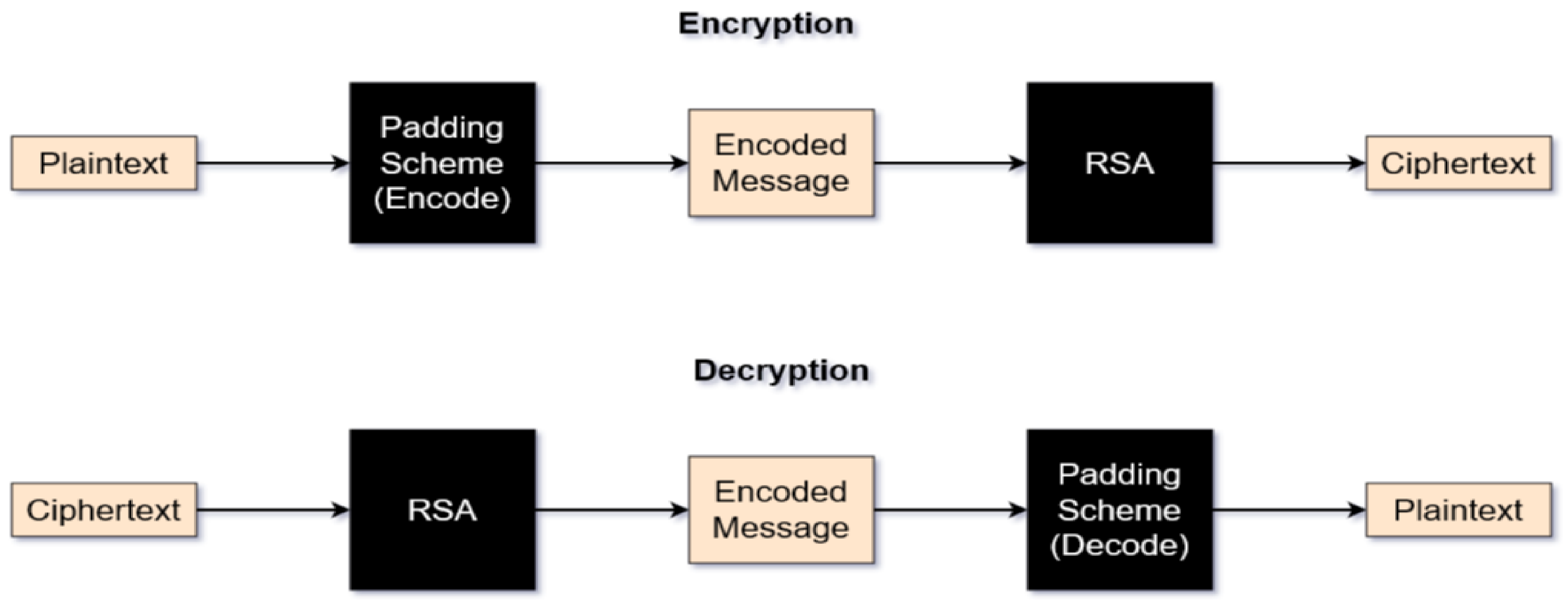

Rivest-Shamir-Adleman (RSA) Public-key Cryptosystem
Key Generation
- Generate primes, p and q
- Compute n = p * q
- Compute Φ(n) = (p-1) (q-1)
- Choose e, such that GCD(e, Φ(n)) = 1
- Compute d, such that d ≡ e^-1 mod ϕ(n)
Public-key = (e, n)
Private-key = (d, n)
Encryption: c = m^e mod n
Decryption: m = c^d mod n

- In the real-world, RSA is implemented with modulus of 2048/4096-bit integers. Thus, RSA is computationally expensive and commonly only used for key transmission/digital signatures.
- RSA is a deterministic cryptosystem, which makes it susceptible chosen-plaintext attacks. To mitigate this RSA is almost always employed along with padding schemes.
- Padding schemes introduce a random component into RSA, thus making the algorithm probabilistic.

GENERATING PRIMES FOR RSA

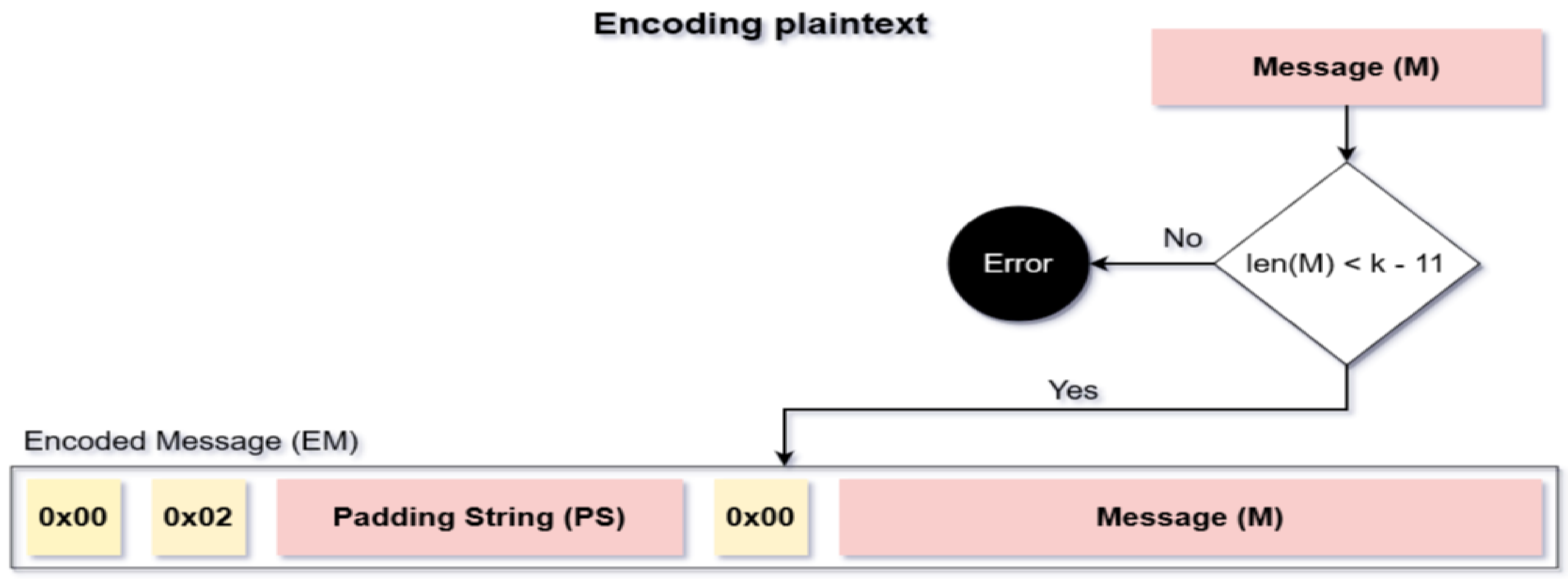
- For a prime number ***p***, if ***p-1*** has many small factors, it is possible to find p given ***p-1*** using **Pollard's *p-1* Algorithm**. This introduces the concept of Safe primes. For a prime ***p***, if ***p-1 = 2 * prime***, then ***p*** is called a **Safe prime**.
- Prime numbers for RSA are generated using cryptographically safe random number generators and often not susceptible to Pollards Algorithm. However, if the prime numbers aren't chosen with care, it could be detrimental to RSA.

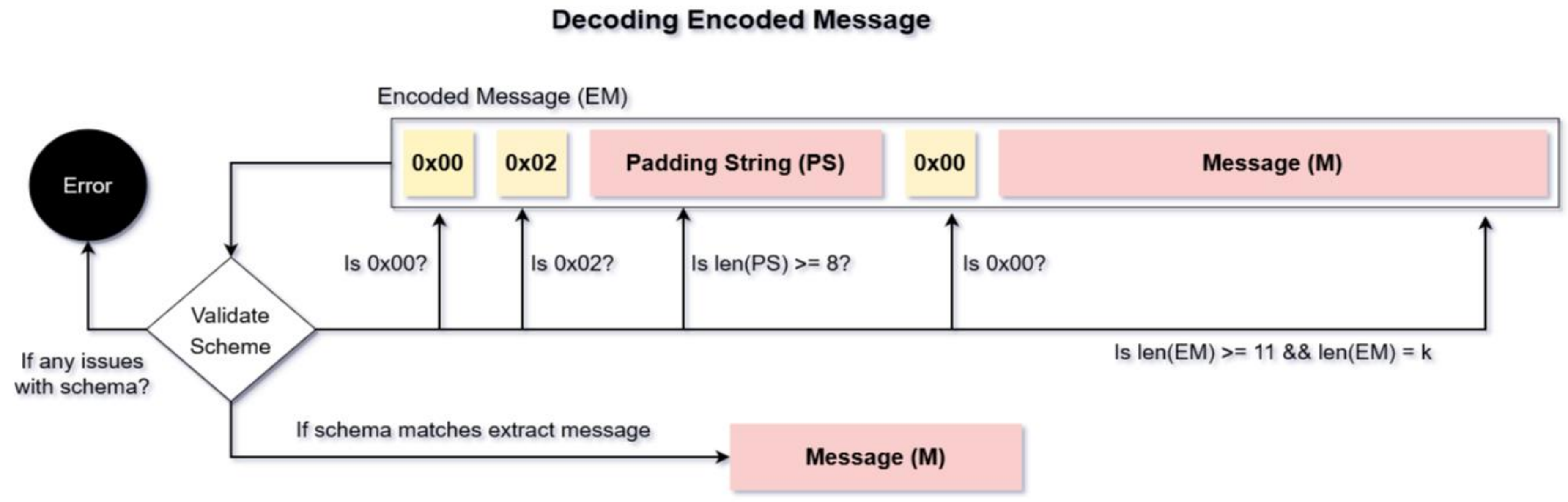
HOW PADDING SCHEME'S FUNCTION



PADDING SCHEMES

Public-Key Cryptography Standard #1 v1.5 (PKCS#1 v1.5)



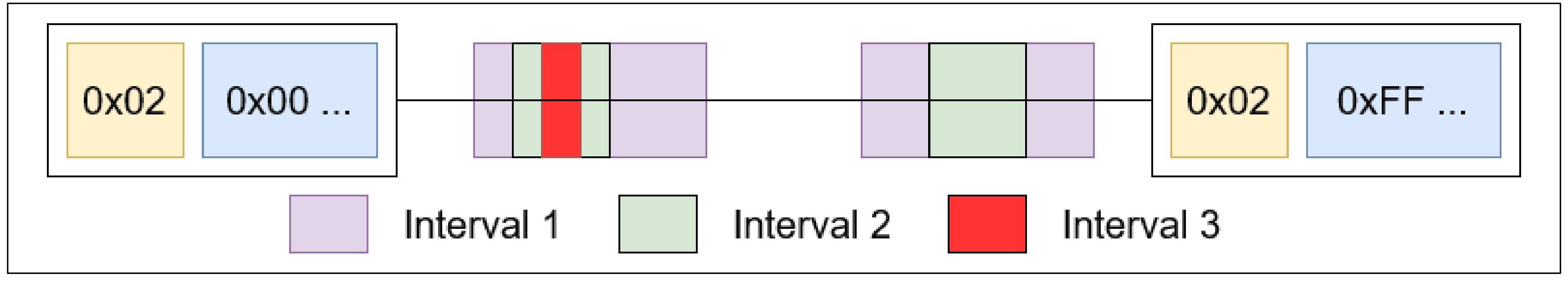


Message Length-Based Analysis

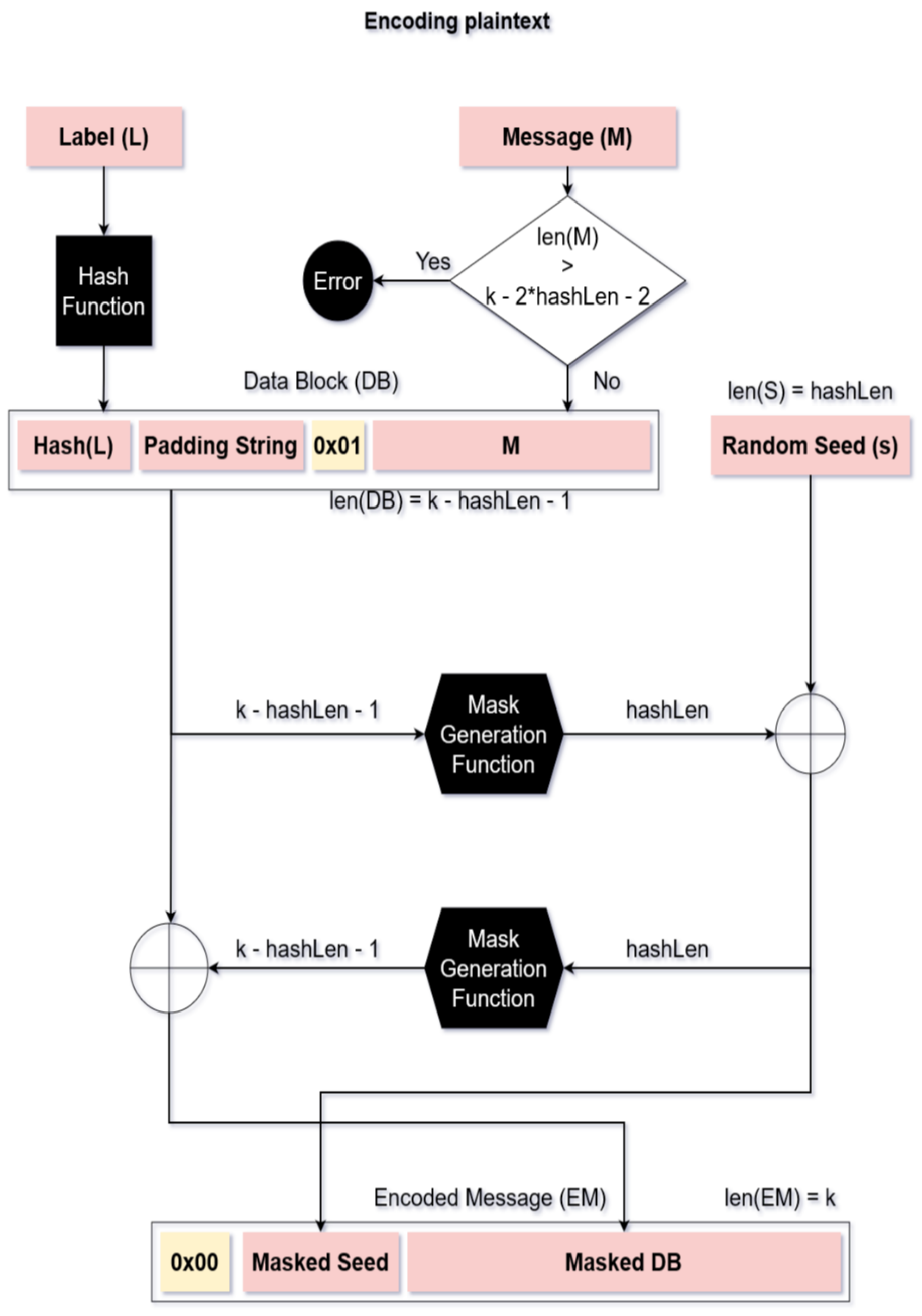
RSA Modulus (bits)	RSA Modulus (bytes)	Max(Len(input message)) bytes	Permutations
1024	128	117	2^{56}
1294	161	150	
1536	192	181	
1626	203	192	
2048	256	245	
4096	512	501	

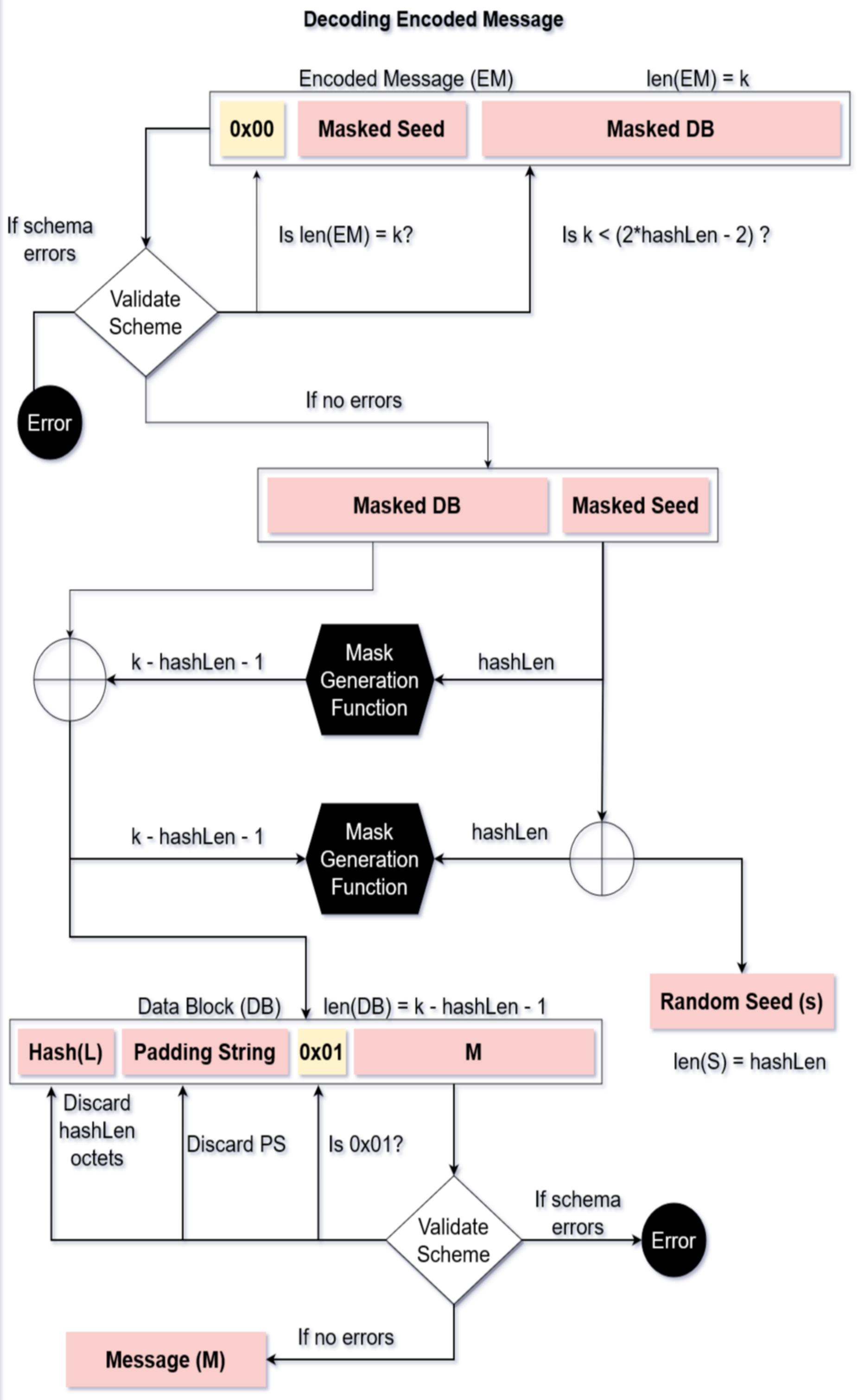
Bleichenbacher's Attack

- This attack exploits the fact that PKCS starts with 0x00 0x02. The attacker must have access to an oracle which confirms if ciphertext is PKCS compliant.



Optimal Asymmetric Padding Encryption (OAEP)



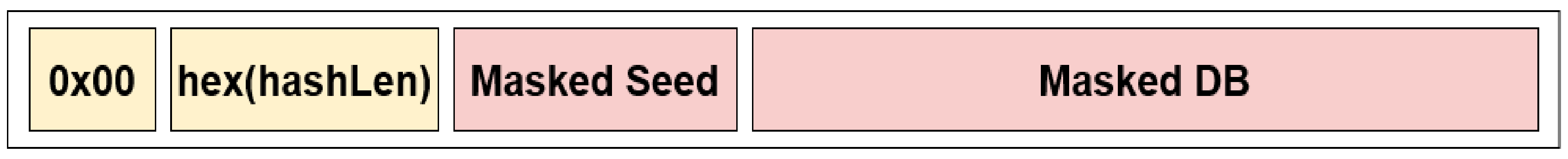


Message Length-Based Analysis

RSA Modulus (bits)	RSA Modulus (bytes)	Max(Len(input message)) (bytes)				Permutations
		SHA3-224	SHA3-256	SHA3-384	SHA3-512	SHA3-n
1024	128	70	62	30	NA	2^n
1294	161	103	95	63	31	
1536	192	134	126	94	62	
1626	203	145	137	105	73	
2048	256	198	190	158	126	
4096	512	454	446	414	382	

SHAKE128/256 with OAEP

- Replace Mask Generation Function with SHAKE125/256.
- Randomly generate hashLen for OAEP as:
  - For SHAKE128: 8 ≤ hashLen ≤ 32
  - For SHAKE256: 32 ≤ hashLen ≤ 64
- While encryption, generate encoded message as shown below. hex(hashLen) is encoded to help with decryption.



REFERENCES

- M. J. Dworkin et al., “SHA-3 standard: Permutation-based hash and extendable-output functions,” 2015.
- C. F. Kerry and C. R. Director, “FIPS PUB 186-4 Federal Information Processing Standards publication Digital Signature Standard (DSS),” 2013.
- K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch, “PKCS# 1: RSA Cryptography Specifications Version 2.2,” Internet Engineering Task Force, vol. 8017, p. 72, 2016.