MINISTRY OF COMMUNICATIONS
AND MULTIMEDIA MALAYSIA

CyberSecurity
MALAYSIA

First edition
October 27, 2020

Technical Report on the Non-Inclusion Cryptographic Algorithms in AKSA MySEAL (AKSA-NICA) v1.0

Reference number:
CD-5-RPT-1220-AKSA_ NICA-V1

REGISTERED OFFICE:

CyberSecurity Malaysia,
Level 7 Tower 1,
Menara Cyber Axis,
Jalan Impact,
63000 Cyberjaya,
Selangor Darul Ehsan, Malaysia
Email: ctr@cybersecurity.my

COPYRIGHT © 2020 CYBERSECURITY MALAYSIA

NO ENDORSEMENT

Products and manufacturers discussed or referred to in this document, if any, are presented for informational purposes only and do not in any way constitute product approval or endorsement by CyberSecurity Malaysia.

TRADEMARKS

All terms mentioned in this document that are known to be trademarks or service marks have been appropriately capitalized. CyberSecurity Malaysia cannot attest to the accuracy of this information. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

DISCLAIMER

This document is for informational purposes only. It represents the current thinking of CyberSecurity Malaysia on the usage of Algoritma Kriptografi Sedia Ada (AKSA) Senarai Algoritma Kriptografi Terpercaya Negara (MySEAL) recommended cryptographic algorithms. It does not establish any rights for any person and is not binding on CyberSecurity Malaysia or the public. The information appearing on this guideline is not intended to provide technical advice to any individual or entity. We urge you to consult with your own organization before taking any action based on information appearing on this guideline or any other documents to which it may be linked.

Contents

1 Introduction

MySEAL, [1] which denotes for National Trusted Cryptographic Algorithm List (*Senarai Algoritma Kriptografi Terpercaya Negara*), is a multiyear Malaysians' project to list trusted cryptographic algorithms. The project is organized by CyberSecurity Malaysia (CSM) and participated by cryptographic academia and experts from various institutions. Its aim is for the list to become the national guideline and security requirement on using cryptographic algorithms in all trusted digital products in Malaysia. The list is divided into two main categories:

1. **AKSA MySEAL.** To list all the recommended **existing** cryptographic algorithms that are already being used in the current cryptographic standards and implementations.
2. **AKBA MySEAL.** To list all the recommended **new** cryptographic algorithms that are developed locally.

Fourty-eight cryptographic algorithms had been evaluated accordingly by its panel of experts to complete the list in AKSA MySEAL. The results of the evaluations were then presented to MySEAL Focus Group before the final list was formed. Only twenty-nine cryptographic algorithms were included in the final list, while the remaining algorithms were excluded due to various technical aspects that cause them to fail the evaluations. However, some of these excluded algorithms are still heavily used in various cryptographic applications and implementations. These algorithms are referred to as Non-Inclusion Cryptographic Algorithms or AKSA-NICA in this document. Three AKSA-NICA are selected and discussed in this document as follows:

- SHA-256;
- RSASSA-PKCS #1 v1.5; and
- RSA-EMSA2.

This document provides justifications on the decisions made to exclude them from AKSA MySEAL. The justifications are to avoid future uncertainty over the usage of these AKSA-NICA. MySEAL hopes that this document will serve as a platform to clarify this ambiguous situation.

Since the evaluations of AKSA MySEAL had been completed at the end of 2017, the justifications to exclude AKSA-NICA from AKSA MySEAL were made based on the publications before the date. Any literature published after the stipulated date that contradicted the justifications made will be taken into account during the next round of AKSA MySEAL evaluation. The evaluation is planned to commence in 2021. These statements also apply to all cryptographic algorithms that had been evaluated in AKSA MySEAL.

1.1 Objective

This document is to provide justifications that result in the non-inclusion of some cryptographic algorithms from AKSA MySEAL. The justifications were based on the outputs gathered from the

[1] https://myseal.cybersecurity.my

AKSA MySEAL panel of experts who evaluated the algorithms. This document also describes the AKSA MySEAL process so that readers can understand the reasons for every justifications made and recognize that the selection of AKSA MySEAL was done with due and thorough processes.

1.2 Scope

This document focuses on selected cryptographic algorithms that failed to be listed in AKSA MySEAL but are still heavily used in various cryptographic applications and implementations. The cryptographic algorithms are as follows:

1. Secure Hash Algorithm 2 or SHA-2 with a 256-bit digest length. This document will refer to it as **SHA-256**.
2. RSA Digital Signature Algorithm with Signature Scheme with Appendix (SSA) for Public-Key Cryptography Standards version 1.5 was released by RSA Laboratories. This document will refer to it as **RSASSA-PKCS #1 v1.5**.
3. RSA Digital Signature Algorithm with encoding method for signature appendix or EMSA2. This document will refer to it as **RSA-EMSA2**.

The details of cryptographic algorithms listed in AKSA MySEAL can be referred to in Guideline on the Usage of Recommended AKSA MySEAL Cryptographic Algorithms (v1.0) document that was released by CyberSecurity Malaysia in 2020.

This document applies to the protection of Official Secret and Official Information of the Malaysian government. This document also applies to Secret Information of non-government Critical National Information Infrastructure (CNII) agencies or organisations. It is inline with the National Cryptography Policy (NCP) described in Section 5.7 of Guideline on the Usage of Recommended AKSA MySEAL Cryptographic Algorithms (v1.0) document.

1.3 Target Audience

This document is intended to provide information to individuals and organisations interested in a specific cryptographic algorithm; however, they failed to find it in the list of AKSA MySEAL recommended cryptographic algorithms. The audience of this document may include, but not limited to:

1. Managers responsible for managing projects that make use of cryptography;
2. Developers implementing, possibly from scratch, AKSA MySEAL recommended algorithms as a cryptographic library;
3. Developers making use of an existing implementation of AKSA MySEAL recommended algorithms in an application;
4. Solution or enterprise architects overseeing cryptography-related projects;
5. Auditors assessing an application, product or service provided by a third party that makes use of AKSA MySEAL recommended algorithms; and
6. Researchers and learners who want to know how to use cryptography properly.

1.4 Document Organisation

This document is organized as follows. Chapter 1 describes this document's motivation, along with its objectives, scopes, and intended audiences. Chapter 2 describes the terms, definitions, and abbreviated terms used in this document. In Chapter 3, the processes towards finalizing trusted cryptographic algorithms into AKSA MySEAL were discussed. Chapter 4 provides a brief description of the cryptographic hash function and its security properties. Justifications to exclude SHA-256 from AKSA list and recommendations on its usage are also given in this chapter. In Chapter 5, a brief description of the digital signature scheme and its security models are provided. Justifications to exclude RSASSA-PKCS #1 v1.5 and RSA-EMSA2 from AKSA list with recommendations by MySEAL regarding both algorithms are also discussed later in this chapter. Finally, this document is summarized in Chapter 6.

2 Terms, definitions, and abbreviated terms

2.1 Terms and definitions

For this document, the following terms and definitions apply:

algorithm
a particular specification of a cryptographic primitive. It consists of computational steps that take one or more inputs and produce one or more outputs. Examples are AES-128 and SHA-256.

bit
one of the two symbols '0' or '1'.

deterministic function
function whose output is the same given the same set of input values.

key agreement
process of establishing a shared secret key between entities in such a way that neither of them can predetermine the value of that key. Note: By predetermining it, it means that neither entity A nor entity B can, in a computationally efficient way, choose a smaller keyspace and force the computed key in the protocol to fall into that keyspace.
(ISO/IEC 11770-3:2015)

may
this word, or the adjective "optional", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option must be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein, an implementation which does include a particular option must be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.).
(IETF RFC 2119)

must
this word, or the terms "required" or "shall", mean that the definition is an absolute requirement of this document.
(IETF RFC 2119)

must not
this phrase, or the phrase “shall not”, means that the definition is an absolute prohibition of this document.
(IETF RFC 2119)

official secret
any document specified in the Schedule (of the Official Secrets Act 1972) and any information and material relating thereto and includes any other official document, information and material as may be classified as ‘Top Secret,’ ‘Secret,’ ‘Confidential,’ or ‘Restricted’, as the case may be, by a Minister, the Menteri Besar or Chief Minister of a State or such public officer appointed under section 2B (of the Act).
(Official Secrets Act 1972)

official information
any data, information and materials related to public service, apart from Official Secret.
(adopted from the National Cryptography Policy)

primitive
a cryptographic process or tool that provides one or more services. Examples are block ciphers and hash functions.

private key
the key of an entity’s asymmetric key pair which should only be used by that entity.
(ISO/IEC 11770-1:1996)

public key
the key of an entity’s asymmetric key pair which can be made public.
(ISO/IEC 11770-1:1996)

probabilistic function
a function whose output does not only depend on given input values, but also on a randomly chosen auxiliary value.

secret information
any data, information, and related materials that need protection and classified by any officer appointed under the rules, circular or laws adopted by the non-government CNII agencies/organisations.
(adopted from the National Cryptography Policy)

secret key
the key used with symmetric cryptographic techniques by a specified set of entities.
(ISO/IEC 11770-3:1999)

should
this word or the adjective “recommended” means that there may exist valid reasons in particular circumstances to ignore a particular item. However the full implications must be understood and carefully weighed before choosing a different course.
(IETF RFC 2119)

should not
this phrase, or the phrase “not recommended” means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this

label.
(IETF RFC 2119)

2.2 Abbreviations

For this document, the following abbreviated terms apply:

AES	Advanced Encryption Standard
AKSA	Existing Cryptographic Algorithms (*Algoritma Kriptografi Sedia Ada*)
ANSI	American National Standards Institute
CNII	Critical National Information Infrastructure
CRYPTREC	Cryptography Research and Evaluation Committees
CSM	CyberSecurity Malaysia
DRBG	deterministic random bit generator
FIPS	Federal Information Processing Standards
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
MySEAL	National Trusted Cryptographic Algorithms List (*Senarai Algoritma Kriptografi Terpercaya Negara*)
NCP	National Cryptography Policy
NESSIE	New European Schemes for Signatures, Integrity and Encryption
NIST	National Institute of Standards and Technology
RFC	Request for Comments
RSA	Rivest-Shamir-Adleman

3 AKSA MySEAL

MySEAL Focus Group members have recommended twenty-nine cryptographic algorithms into the trusted cryptographic algorithms list, AKSA MySEAL. This chapter describes all stages that lead to the final list of AKSA MySEAL. The purposes of this chapter are:

1. To provide reasons for processes towards non-inclusion decisions of some algorithms described in Chapters 4 and 5.
2. To show that the non-inclusion actions have been decided after collective judgements based on objective evaluations.
3. To give some disclosure on internal processes involved in the finalization of AKSA MySEAL.

3.1 Development and finalization of evaluation criteria

The first step in the AKSA MySEAL project was to develop the submission and evaluation criteria for utilization in the next phases. The developed submission criteria were used in Phase 1 evaluation while the developed evaluation criteria were used in Phase 2 evaluation. A total of forty-eight cryptographic algorithms were considered as initial candidates for AKSA MySEAL. These algorithms were categorized into six cryptographic primitives as follows:

a. Symmetric Block Cipher;

b. Symmetric Stream Cipher;

c. Asymmetric Cryptographic; which can be categorized to:

   i. Digital Signature Scheme;
   ii. Asymmetric Encryption Scheme; and
   iii. Key Agreement Scheme.

d. Cryptographic Hash Function;

e. Prime Number Generators; and

f. Deterministic Random Bit Generator.

Most of the considered algorithms were also reported in the following standardization efforts and other cryptographic algorithm listing projects:

i. **NIST FIPS.** The publications of Federal Information Processing Standards (FIPS) by the National Institute of Standards and Technology of US are intended to protect non-national security federal information systems. The publications also have been adopted and used by non-federal government organizations and private sector organizations due to its definitive and competency to the latest development in the subject matter.

ii. **ISO/IEC.** This voluntary, consensus-based, market-relevant standardisation is supervised by its technical committee, ISO/IEC JTC 1/SC 27. It is to include generic methods, techniques, and guidelines to cryptographic and other security mechanisms, including but not limited to mechanisms for protecting the accountability, availability, integrity, and confidentiality of information

iii. **CRYPTREC.** Established by the Japanese government in 2000, Cryptography Research and Evaluation Committees or CRYPTREC's main aim is to evaluate and monitor the security of e-Government recommended ciphers and to examine the establishment of evaluation criteria for cryptographic modules. The committee actively updated its guidelines and recommendations up until 2018.

iv. **Public Key Cryptography Standards (PKCS).** The standards focus on the usage of RSA cryptosystem and the other cryptographic applications that are using the cryptosystem. It was published by RSA Security LLC and is currently maintained by the company, which retained control.

v. **NESSIE.** This research project collaborated among the cryptographic academia in Europe to put forward a portfolio of strong cryptographic primitives obtained after an open call and evaluated using a transparent and open process. It has been inactive for many years, but the industries are still using some of the algorithms approved by this project.

vi. **ANSI X9.31.** This is specifications for using digital signatures in the financial services industry, which formed a document entitled "Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)."

vii. **e-STREAM.** The European Network of Excellence in Cryptology or ECRYPT under the European Union initiative organised this project to select suitable stream ciphers categorised into two profiles: software and hardware.

3.2 Phase 1 evaluation

In Phase 1, all the cryptographic algorithms were evaluated based on the following criteria:

1. **Security.** The primary function of cryptographic algorithm is to secure communications. Thus, the soundness of security of an algorithm was the most critical aspect to be evaluated. This criterion carried the highest weightage in the evaluation.
2. **Implementation and performance.** This criterion was evaluated based on the existing implementation(s) of the algorithm and the availability of its performance report(s).
3. **Supported parameters.** In this criterion, each parameter used by the algorithm must be adequately explained and provided in the literature.
4. **Others.** Unique criterion based on the primitives and schemes was also evaluated. For example, the existence of mathematical proof of correctness in asymmetric cryptographic schemes is an essential criterion during the evaluation of the scheme.

A scoring matrix was developed to quantify the evaluation for each primitive or scheme. Each algorithm was scored either 0 if it failed, or 1, if it satisfied each criterion. Each algorithm was then checked if it passed the minimum marks set for each primitive or scheme. Different cryptographic primitive or scheme was evaluated by different teams that consisted of experts for the primitive or scheme.

3.3 Short-listing algorithms which pass evaluation Phase 1

Cryptographic algorithms that achieved a minimum score of 75% were short-listed for further evaluation in Phase 2. Out of forty-eight algorithms evaluated in Phase 1, six algorithms scored below the minimum percent, thus were not recommended the second phase of the evaluation[2]. For example, RSASSA-PKCS-v1.5, which is discussed in Section 5.2, did not meet the minimum criteria stated in the scoring matrix and, therefore, was dropped for Phase 2 evaluation.

3.4 Phase 2 evaluation

In Phase 2, a stricter and more rigorous evaluation was conducted. The evaluation criteria for the cryptographic algorithms were also standardized into four major criteria. In each criterion, more detailed and meticulous sub-criteria were introduced, each was refined based on the primitives and schemes. The major criteria are:

1. **Security.** The evaluation of the security for cryptographic algorithms in Phase 2 still contributed the largest percentage of the evaluation's total score. Unlike the earlier evaluation, which only required the existence of past attacks, in this phase, the evaluation focused on the magnitude of the attack and its consequences to the security of the cryptographic algorithms. For example, a higher number of rounds susceptible to attack would result in a lower score in the evaluation for cryptographic hash functions.

[2] refer to Appendix A for the full list of algorithms that are excluded in Phase 1 evaluation.

2. **Cost, performance, and implementation characteristics.** This criterion mostly examined the computational speed and complexity in the algorithm's implementations and whether its flexibility was able to improve it. For primitives with minimal computational complexity, namely prime number generators and deterministic random bit generators, an acceptance index that discussed the deployment rate of the algorithm in software or hardware was considered.
3. **Soundness of design.** The justification for the design of the algorithm was determined in this evaluation.
4. **Maturity.** This criterion evaluated the general acceptance of the algorithm by the cryptographic and developer communities. It also included the number of years since published, the number of citations for the paper, the number of existing protocols that implement the scheme, and the number of cryptographic libraries available that support the scheme's implementation.

A scoring matrix for Phase 2 was designed to give a weightage on each sub-criterion. The range of scores from 0 to 5 gave values for the matrix. It should be noted here that the evaluation for cryptographic primitives that support more than one key or digest length, where relevant, was performed separately. The primitives involved were block ciphers, stream ciphers, hash functions, and DRBGs. Therefore, there may be cases where an algorithm was approved for use only with a selected key or digest lengths.

3.5 Short-listing algorithms which pass evaluation Phase 2

In Phase 2 evaluation, nineteen algorithms and/or its variants were not recommended to be included in AKSA MySEAL list[3] [4]. For example, SHA-256 and RSA EMSA2, which are discussed in Sections 4.2 and 5.3 respectively, were excluded from the list.

4 Cryptographic Hash Functions

4.1 Overview of Cryptographic Hash Functions

Hash function is an algorithm that takes data with an arbitrary size (called "message") as its input to produce a fixed length of outputs (called "message digest") that are organized as arrays of bit. It is a one-way function that practically takes an infeasible amount of time to find the value of its inverse function. Hash function has been used in many applications to achieve cryptographic properties such as:

- Generating a unique signature for verification;
- Comparison of two or more message digests to check for data integrity;
- Storage and verification form for passwords; and many others.

There are three main properties of hash function to resist attacks against the algorithm:

[3] exclusion of certain variants of an algorithm did not imply other variants of the same algorithm should also be excluded. For example, only SHA-224 and SHA-256 variants from SHA-2 family were not recommended while other variants of SHA-2 were recommended

[4] refer to Appendix B for the full list of algorithms that are excluded in Phase 2 evaluation

(a) **preimage resistance.** Given a message digest $H(m)$, it should be computationally infeasible for an adversary to find m. This propoerty is connected to the one-way function that should be employed by any secure cryptographic hash function.

(b) **Second preimage resistance.** Given a message m_1, it should be computationally infeasible for an adversary to find m_2 such that $H(m_1) = H(m_2)$ where $m_1 \neq m_2$.

(c) **Collision resistance.** Any cryptographic hash function employs this resistance if it is computationally infeasible for an adversary to find $H(m_1) = H(m_2)$ where $m_1 \neq m_2$.

An attack that is closely related to (c) is called **pseudo-collision attack**. It manipulates Merkle-Damgård construction [1] used in a one-way internal compression function of a hash function. This attack gives insights into a possible path to full-collision attacks.

Another type of attack is **distinguishing attack**, and it assumes that the attackers can distinguish a hash function from a random function.

All of these attacks are considered in the evaluation of cryptographic hash function in AKSA MYSEAL.

4.1.1 On the Evaluation of Cryptographic Hash Functions in AKSA MySEAL

From Sections 3.2 and 3.4, it is known that security is the most crucial criterion to be evaluated in Phases 1 and 2 evaluations of AKSA MySEAL. The following are the sub-criteria of security considered in Phase 2 evaluation:

(a) The number of attacks or cryptanalysis published that threaten the security properties of the hash function. All types of attacks, including collision, preimage, second preimage, and distinguishing attacks, are considered. In other words, this sub-criterion looks into the depth of analysis performed by the cryptographic community on hash functions; and

(b) The number of rounds attacked refers to the largest number of rounds of the hash function (or its underlying components) that have successfully being attacked using any of the cryptanalytic techniques highlighted in Section 4.1. It also can be computed into a security margin using the following formula:

$$\text{security margin} = \frac{\text{total number of rounds} - \text{number of rounds attacked}}{\text{total number of rounds}}$$

In most cases, the security evaluations of hash functions in AKSA MySEAL focus on the resistance against preimage and collision attacks (since successful collision also implies the existence of second preimage). Note that the evaluation results do not necessarily yield definite rules for selecting or discarding any of the algorithms. However, the results do provide a reasonably good indication of security.

Other criteria considered in Phase 2 evaluation relatively share the same description with the evaluation criteria discussed in Section 3.4 which are:

- Cost, performance and implementation characteristics;
- Soundness of design; and
- Maturity.

4.2 SHA-256

4.2.1 Overview

The Secure Hash Algorithm (SHA) refers to the families of hash functions published by NIST in the FIPS 180 standard. The first of the family, SHA-0, was designed by the NSA and published by NIST in 1993. Two years later, SHA-0 was found to be weak and replaced by SHA-1 that differs slightly from SHA-0. In 2001, the SHA-2 family of hash functions was introduced and added to the FIPS 180-2 standard. It contains notable revisions from its predecessor, SHA-1, and comprises four hash function algorithms. In 2012, the standard was revised as FIPS 180-4 to include two additional algorithms.

The SHA-2 family consists of six algorithms to connote the different digest lengths: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. The last two algorithms in this list can be regarded as truncated versions of SHA-512. Although the digest lengths of these algorithms are identical to SHA-224 and SHA-256, the SHA-512/t variants can handle a larger block[5] and subsequently, a significantly larger message space. All SHA-2 algorithms are based on the MD structure. The underlying compression function is an amalgamation of bitwise AND, XOR, NOT, and rotation operations.

SHA-2 is listed in the following standards and cryptographic algorithm listing projects:

- NIST FIPS PUB 180-4 – Secure Hash Standard (SHS).
- ISO/IEC 10118-3:2004 Hash Functions, Part 3: Dedicated Hash Functions.
- NESSIE Collision-Resistant Hash Functions.
- CRYPTREC e-Government Recommended Ciphers List.

Apart from NIST FIPS PUB 180-4, the SHA-2 family defined in the above standards are SHA-256, SHA-384, and SHA-512 only. At the time of writing, the ISO/IEC 10118-3 standard is being drafted to incorporate SHA-224, SHA-512/224, and SHA-512/256.

4.2.2 Results of SHA-256 Evaluation

Security, as the main evaluation criterion, considers cryptanalysis work that directly threatens the security properties of a hash function.

AKSA MySEAL's Phase 2 evaluation report showed that most of the evaluated cryptographic hash functions have yet to be broken or threatened by any practical attack. However, only some of the hash functions have sufficient proof relating their security to their compression function's ideal properties.

In terms of security evaluation, SHA-256 has been the target of most cryptanalytic research, and it has been analysed against many types of collision attacks [2, 3, 4, 5, 6, 7, 8]. The reason why SHA-256 has been the most targeted attack hash function is that it is one of the most popularly used for a wide variety of applications and protocols.

Table 4.1 shows a more detailed cryptanalytic result, i.e. number of rounds being broken by cryptanalytic attacks for SHA-256. Note that the results cover all the applicable attacks on SHA-256,

[5] The length of message chunks that can be processed at a time.

including distinguisher attacks. From the MySEAL evaluators' point of view, although the intensity of cryptanalysis is much higher on SHA-256, the primitive is still secure in practical (real-time) attacks. However, these findings indicate that the security margin of SHA-256 is insufficient. In other words, SHA-256 is not conditionally/computationally secure, whereby there exist attacks (faster than an exhaustive search) for up to 81% of SHA-256's rounds. NIST also had performed testing on SHA-256 that can be referred to in more details from [9].

Table 4.1: Summary of the best attacks against SHA-256

Rounds	Security Margin	Time Complexity	Attack	Year
52/64	0.19	$2^{127.5}$	Pseudo-collision [6]	2012
52/64	0.19	2^{255}	preimage [5]	2012
51/64	0.20	2^{218}	Distinguisher [10]	2016
47/64	0.27	$> 2^{85}$	Collision [3]	2011
45/64	0.30	$2^{255.5}$	preimage [2]	2009

In terms of the algorithm's computational efficiency and flexibility, SHA-256 generally requires fewer cycles per byte [11] than its benchmarked algorithm, KECCAK [12], and has a higher flexibility. The flexibility of SHA-256 sets out two factors, namely:

- SHA-256 can accommodate additional digest size.
- SHA-256 can be implemented in a wide variety of platforms/applications.

Hence, SHA-256 was given a higher score for accommodating other SHA-2 family digest sizes (224, 384). Its design, which is based on the Merkle-Damgård structure, is one that has been well-studied and well-accepted.

In terms of maturity, SHA-256 is widely accepted and used by the cryptographic and developer communities. Thus, for the remaining three criteria that were evaluated in Phase 2 evaluation, SHA-256 received high scores than the passing marks of each criterion.

Justification for the Non-Inclusion

SHA-256 was not recommended in MySEAL AKSA in 2017 due to its low security margin of 0.19 against an attack in 2012 [6]. The ability to conduct this attack is due to the internal compression function in SHA-256, which utilizes Merkle-Damgård construction. By manipulating the construction, an adversary can convert a meet-in-the-middle preimage attack to a pseudo-collision attack, which can identify collisions for up to 52/64 rounds or 81% of rounds in SHA-256, based on Table 4.1. Since the number of rounds is one of the sub-criteria evaluated in the security criterion, which brings the largest percentage of the total score in Phase 2 evaluation, the algorithm did not meet the minimum security margin criterion. Hence it could not be listed in AKSA MySEAL.

The consequences of this result can be anticipated in an almost similar situation that occurred with SHA-1. The algorithm had a security margin of 0.34 in January 2005 [13], i.e. cryptanalytic attacks could identify collisions for up to 66% of its rounds. However, a breakthrough in cryptanalysis was discovered in February 2005, rendering SHA-1 insecure [14]. It led to the immediate non-recommendation of SHA-1 in practical applications in 2006 [15], and the call for proposals for SHA-3 in 2007 [16].

As in most cryptographic primitives and schemes, the significant factors of the advancement of an attack depend on the rise of computational power and the potentiality of a novel attacking technique.

If these factors are considered and applied to this situation, SHA-256 may no longer be recommended in other cryptographic standards and implementations as its predecessors, SHA-0, and SHA-1 in the near future.

4.2.3 Recommendation from MySEAL

From the previous section, it has been presented that SHA-256 has a low security margin against pseudo-collision attack. For that, SHA-256 may not be used anymore, especially in the upcoming cryptographic applications. Hash functions with higher security margins should instead be considered. As such, AKSA MySEAL panel of experts recommended the following alternative hash functions that have higher security margins:

- SHA-384, -512, -512/224, -512/256
- All SHA-3 variants

However, it must be noted that all SHA-2 variants have Merkle-Damgård construction in its compression function. On the other hand, SHA-3 does not share this same property. Thus, practitioners **should** conduct a cost-effective analysis to arbitrate the cost of migrating to the same family of SHA-2 compared to SHA-3 since the same type of attacks on SHA-256 will become more effective over time due to advancements in computing capability.

5 Asymmetry Cryptography: Digital Signature Schemes

5.1 Overview of Digital Signature Scheme

Digital signature scheme is a group of mathematical algorithms used to generate a digital signature of an entity that enables the signature's recipient to verify it. Using a concept from asymmetric cryptography, it also frequently utilizes cryptographic hash functions to satisfy its security goals.

The adversary trying to break a digital signature scheme is called a forger.

The following attack models describe the adversarial capabilities of a forger when attacking digital signatures:

1. **Key only attack.** In this model, a forger is given only the public verification key.
2. **Known message attack.** In this model, a forger is given valid signatures of some pre-selected arbitrary messages generated by the scheme.
3. **Non-adaptive chosen message attack.** In this model, a forger is given the valid signatures of arbitrary messages generated by the scheme, which have been pre-selected by the forger before the forger knows the value of public key.
4. **Adaptive chosen message attack.** In this model, a forger is given the capability to obtain valid signatures of arbitrary messages chosen by the forger at any time.

The strength of the adversary increases from the top to the bottom of the list.

The following adversarial goals describe what the forger wants to achieve in breaking a digital signature scheme:

1. **Total break.** This condition occurs when the attack results in the forger obtaining the signing key.
2. **Universal forgery.** This condition occurs when the attack results in the forger being able to forge any signature from any random message.
3. **Selective forgery.** This condition occurs when the attack results in the forger being able to forge a signature from a message of adversary's choice.
4. **Existential forgery.** This condition occurs when the attack produces valid pairs of signature and messages which were not already queried before by the forger.

As with the prior list, the notion of security grows from the top to the bottom of the list.

Combined, the two lists above fully describe the scheme's security model, which includes adversarial capabilities and adversarial goals of a forger. For AKSA MySEAL, only digital signature schemes that fulfil the minimum requirement of being secure against existential unforgeability under chosen message attacks (EUF-CMA) were included[6] are considered secure.

5.1.1 Security Proofs

Security proofs in cryptography are defined as a demonstration to prove a cryptographic scheme secure against a defined security model assuming the intractability of specific hard mathematical problems. An algorithm is said to achieve a security proof if it passes the a security model's requirements while giving the adversary with access to the algorithm and reasonable computational powers. Based on the security models described in the previous section, the security proof is developed from an experiment or 'game' between an adversary and challenger. A proving technique would be to reduce a successful adversary breaking the scheme to one that solves an intractable mathematical problem, thereby achieving a contradiction. It is because no such algorithms exist to solve the intractability of the mathematical problem, thereby rendering the scheme secure by showing the non-existence of such adversaries. Security bounds are then determined by how tight or loose this adversary is modelled against the intractable mathematical problem and directly affecting the key sizes required to secure the cryptographic scheme.

For AKSA MYSEAL, the evaluation only considers the existence and correctness of such proofs, but not the security bounds within the proof.

The security model can be updated with a stronger notion of security if the algorithm is proven secure when more advantages are given to the adversary in the game; in the form of oracle queries. For example, stronger EUF-CMA security model called SEUF-CMA is achieved if the adversary is allowed to query the signing oracle for a signature on the challenge message, while key only attacker has no access to a signing oracle at all.

[6] EUF-CMA security model is not the strongest security model. Refer to Section 5.1.1 for more details

5.1.2 On the Evaluation of Digital Signature Schemes in AKSA MySEAL

From Sections 3.2 and 3.4, it was known that security is the most important criteria to be evaluated in Phases 1 and 2 evaluations of AKSA MySEAL. The following are the sub-criteria considered in Phase 2 evaluation:

(a) The assumptions on the scheme, mainly the difficulty of breaking the hard mathematical problem such as RSA, factoring or solving discrete logarithms; and

(b) The correctness of the security model and its proof must be in line with the assumptions made in (a).

For digital signature schemes, security against EUF-CMA adversary is a minimum requirement. The schemes were regarded higher in terms of security if they are proven secure against strong EUF-CMA (SEUF-CMA). Schemes that are secure against a stronger notion of security, preferably without the use of random oracles, were also preferred. Maximum evaluation scores were given to schemes proven secure against side-channel attacks and other more theoretical opponents such as related-key attacks and exploitation of weak randomness during key generation.

Other criteria considered in Phase 2 evaluation shared the same descriptions with the evaluation criteria discussed in Section 3.4, which were:

- Cost, performance and implementation characteristics;
- Soundness of design; and
- Maturity.

5.2 RSASSA-PKCS-v1.5

5.2.1 Overview

The RSASSA-PKCS #1 v1.5 digital signature is a hash-and-sign digital signature based on the RSA assumption to be proposed for standard usage. Version 1.5 was published as public with Requests for Comments (RFC) in March 1998. The latest version 2.2 is RFC8017, which was published in November 2016 [17].

During the initial evaluation of MySEAL AKSA, RSASSA-PKCS #1 v1.5 was eliminated from Phase 1 following no evidence of a security proof, and mainly due to a note in Section 8.2 of [17] that advises on the deprecation of the scheme in favour of RSA-PSS which is provable secure.

RSASSA-PKCS #1 v1.5 is included in the following standards and cryptographic algorithm listing projects:

- NIST FIPS PUB 186-4 – Digital Signature Standard (DSS) [18].
- NIST FIPS PUB 186-5 – Digital Signature Standard (DSS) draft [19], with the condition of only approved hash functions are to be used.

- ISO/IEC 14888-2:2008/AC1:2015 – Information technology Security techniques Digital signatures with appendix [20].
- CRYPTREC e-Government Recommended Ciphers List. [21]
- Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 1.5.
- TLS 1.3, but only for backward-compatibility reasons [22].

RSASSA-PKCS #1 v1.5 is not included in NESSIE [23].

Details of RSASSA-PKCS #1 v1.5

RSASSA-PKCS #1 v1.5 signatures are RSA signatures with specialized padding of the message to be signed. As with all RSA cryptosystems, a public, private key pair is produced by $KeyGen$ algorithm, which takes in the security parameter and produces $N = pq$ where p, q are large prime numbers. e is selected in $\phi(N)$ and its inverse d is calculated where $ed \cong 1 \mod \phi(N)$. N, e are the public keys while p, q, d are private.

The signature is generated by signing the message M with d the private key, and verification is done by verifying using e.

The main difference of RSASSA-PKCS #1 v1.5 compared to other digital signature schemes is the padding used to generate the message M. Padding takes a hexadecimal format of 0x00$||BT||PS||$0x00$||D$, where:

1. BT is the type of block, and 0x01 is for signatures.
2. D is the encoding of the message, which for signatures is the hash of the message prefixed with the hash id ID_H
3. PS is the padded string to pad the message to n bits of length and is fixed to 0x$FF \ldots FF$. for signatures.

In summary, $M =$ 0x00$||$0x01$||$0x$FF \ldots FF||$0x00$||ID_H||H(m)$ where m is the actual message to be signed.

5.2.2 Results of RSASSA-PKCS #1 v1.5 Evaluation

RSASSA-PKCS #1 v1.5 failed to pass Phase 1 evaluation. The following description describes its results:

The main security sub-criterion in Phase 1 evaluation is the availability of security proof, which is missing from RSASSA-PKCS #1 v1.5. The missing proof posed a problem in its evaluation since the security criterion carries the highest weight in the evaluation. In terms of cost, performance, and implementation characteristics, RSASSA-PKCS #1 v1.5 digital signature scheme ranks well due to its minimal number of operations, which practically only involves hashing the message and then signing it with the RSA exponent, which applies only one exponentiation. In terms of signature size, it is equivalent to the RSA modulus. It is on par with RSA-EMSA2 signatures as it involves the same operations, just different padding. Operations-wise, only one exponentiation is involved as the rest, such as padding and hashing, are considered negligible. Its design enables it to achieve fast signing and verification speeds and small signature size while being provable secure in terms of design principles.

Justification for the Non-Inclusion

RSASSA-PKCS #1 v1.5 was not included in Phase 2 review due to the following reasons:

(a) **Lacking in security proof**: To the date of publication of MySEAL AKSA Phase 1, there was no documentation or academic paper of a security proof reducing the hardness of an existential forgery under chosen message attack to solving factoring or breaking RSA. Therefore, the security of RSASSA-PKCS #1 v1.5 is unknown.

(b) **Note of obsoletion in RFC3447 Section 8.2**: In the last sentence of that note in Section 8.2 of [24], it is recommended that while there are no known attacks that are successful yet, a transition should happen to RSA-PSS as a precaution against future developments of attacks, partly due to the deterministic padding described in Section 5.2.1. For the reader's convenience, the RSA-PSS uses probabilistic padding instead.

(c) **Non-availability of proof of correctness.** The original document proposing RSASSA-PKCS #1 v1.5 did not include its proof of correctness. Therefore, this results in a low evaluation score in Phase 1 evaluation.

(d) **Non-availability of supported parameters.** As mentioned in Section 3.2, a description of supported parameters was an important criterion in Phase 1 evaluation. However, the initial proposal of RSASSA-PKCS #1 v1.5 lacks the recommended minimum key length needed to achieve the security level equivalence of AES-128 and test vectors and efficiency and complexity analysis.

However, we note that items (c) and (d) can be inferred from NIST and ISO documents listed from Section 5.2.1.

5.2.3 Recommendation from MySEAL

Phase 1 evaluation of AKSA MySEAL was completed at the end of 2017 and decided to exclude RSASSA-PKCS #1 v1.5 from further evaluation due to missing security proof for the algorithm. However, recent work by [25] has proved EUF-CMA security proof for RSASSA-PKCS #1 v1.5. Thus, it should be noted that the algorithm **should** be evaluated in the next cycle of AKSA MySEAL evaluation. For the time being, industry practitioners **should** migrate to RSA-PSS [26], which is both probabilistic and efficient.

Industry practitioners **should** be extremely cautious when selecting RSA parameters to avoid Coppersmith [27] and Wiener's [28] attacks. Other forms of attacks on RSA can also be found in the survey by Boneh[29]. Also, hash functions and implementation of signature verification errors should be done according to standards specifications, such as NIST or ISO documents.

5.3 RSA-EMSA2

5.3.1 Overview

The RSA Digital Signature Algorithm with EMSA2 (RSA-EMSA2) encoding is an RSA signature with a hash-based message encoding designed for ANSI X9.31. In PKCS #1 v2.1, the encoding method is called EMSA2.

RSA-EMSA2 is included in the following standards and cryptographic algorithm listing projects:

- ANSI X9.31: Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry [18].
- IEEE P1363: IEEE Standard Specifications for Public Key Cryptography (2000) [20].
- NIST FIPS 186-4: NIST Digital Signature Standards [21]
- Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.

RSA-EMSA2 is not included in NESSIE [23] and will not be featured in TLS 1.3 [22]. It is also removed from the NIST FIPS 186-5 draft [19], dated 2019.

Details of RSA-EMSA2

Similar to RSASSA-PKCS #1 v1.5 signatures. RSA-EMSA2, or sometimes known as the ANSI X9.31 signature, are RSA signatures with specialized padding of message to be signed. As with all RSA cryptosystems, a public, private key pair is produced by $KeyGen$ algorithm, which takes in the security parameter and produces $N = pq$ where p, q are large prime numbers. e is selected in $\phi(N)$ and its inverse d is calculated where $ed \equiv 1 \mod \phi(N)$. N, e are the public keys while p, q, d are private.

The signature is generated by signing the message M with d the private key, and verification is done using the verification key e.

The padding method used by RSA-EMSA2 is quite similar to that of RSASSA-PKCS #1 v1.5, except, rather than being right-justified, there is a 2-byte trailer with a fixed value that follows the hash value. RSA-EMSA2 uses the following form of 0x06$||PS||D$, where:

1. PS is the padded string to pad the message to n bits of length and is fixed to 0x$BB \ldots BA$. for signatures.
2. D is the encoding of the message, which for signatures is the hash of the message suffixed with the hash id ID_H. For $SHA-1$ this value is 0x33CC

In summary, $M =$ 0x6$||$0x$BB \ldots BA||H(m)||ID_H$ where m is the actual message to be signed.

5.3.2 Results of RSA-EMSA2 Evaluation

RSA-EMSA2 failed to pass Phase 2 evaluation. The following description describes its results:

In terms of cost, performance, and implementation characteristics, RSA-EMSA2 scores well due to its minimal number of operations, which essentially only involves hashing the message and then signing it with the RSA exponent, which involves only one exponentiation. In terms of signature size, it is equivalent to the RSA modulus. It is on par with RSA-FDH signatures as it applies the same operations, just different padding. Operations-wise, only one exponentiation is involved as the rest, such as padding and hashing, are considered negligible.

RSA-EMSA2 also has a high maturity in the cryptographic and developer communities based on the number of years since publication, number of citations on anchor papers based on Google Scholar citation, number of protocols implementing the algorithm, and the number of cryptographic libraries implementing the algorithm. In terms of design principles, the P1363 document [30] and the ANSI X9.31 [31] document shows some vague descriptions that enable RSA-EMSA2 to achieve fast signing and verification speeds and small signature size.

Justification for the Non-Inclusion

RSA-EMSA2 did not pass Phase 2 evaluation due to the following reasons:

(a) **Lacking in security proof**: To the date of publication of MySEAL AKSA Phase 1, there was no documentation or academic paper of a security proof reducing the hardness of an existential forgery under chosen message attack to solving factoring or breaking RSA. Therefore, the security of RSA-EMSA2 is unknown.

(b) **Deterministic Padding**: Similar to RSASSA-PKCS #1 v1.5, RSA-EMSA2 utilizes a deterministic padding methodology. While no known attacks are established, following similar reasoning, the signature should be transitioned towards RSA-PSS, which utilizes probabilistic padding.

(c) **Removal from Standards**: As mentioned in section 5.3.1, RSA-EMSA2 is not included in NIST FIPS 186-5 draft, the latest NIST document [19]. It is also not featured in TLS 1.3 as according to [22]. Therefore, the evaluators of MySEAL AKSA are following suit and conclude that there is no reason to pursue the inclusion of RSA-EMSA2 in MySEAL.

(d) **Obsolete Hash Function support**: RSA-EMSA2 only supports SHA-1 and RIPEMD-160 hash functions. Both hash functions are known to be broken; SHA-1 is broken in [14] while RIPEMD-160 is broken in [32].

(e) **Inactive-reserved Status from P1363**:P1363 website [30] states that the status inactive-reserved is for all standards except for identity-based signatures. This status also includes RSA-EMSA2.

5.3.3 Recommendation from MySEAL

Due to not having a valid security reduction that proves the security of the scheme (refer to Section 5.1.1), and the deterministic padding mechanism that raises concerns on the security, AKSA MySEAL evaluators conclude that RSA-EMSA2 **should not** be included in the trusted digital signatures of MySEAL AKSA list. Therefore, industry practitioners **should** migrate to RSA-PSS [26], which is both probabilistic and efficient.

Industry practitioners **should** extremely be cautious when selecting RSA parameters to avoid Coppersmith [27] and Wiener's [28] attacks. Other forms of attacks on RSA can also be found in the survey by Boneh [29]. Also, hash functions and implementation of signature verification errors should be done according to standards specifications such as NIST or ISO documents.

6 Summary

AKSA Non-Inclusion Cryptographic Algorithms or AKSA-NICA is referred to cryptographic algorithms that failed to pass either Phase 1 or Phase 2 evaluation in AKSA MySEAL but are still being used heavily in cryptographic implementations and applications. This document discussed three such algorithms.

The first algorithm, SHA-256, failed the Phase 2 evaluation of AKSA MySEAL due to its low security margin of 0.19 against an attack that manipulated the construction of its internal compression function. The second algorithm, RSASSA-PKCS #1 v1.5, failed the Phase 1 evaluation of AKSA MySEAL mainly because it lacks security proof and several other technical reasons. The third algorithm, RSA-EMSA2, also shared the same justification for its exclusion due to its missing security proof that caused it to obtain low evaluation scores in Phase 2 of evaluation.

It should be noted here that the evaluations conducted during AKSA MySEAL **DO NOT** imply that the non-inclusion algorithms are not secure or should be immediately refrained from current use. It only means that the algorithms failed to meet certain evaluation criteria set up by AKSA MySEAL. Thus, the algorithms cannot be included in the list of trusted cryptographic algorithms of AKSA MySEAL.

Appendix A

List of cryptographic algorithms that failed Phase 1 evaluation of AKSA MySEAL

Block cipher

CIPHERUNICORN-E

Stream cipher

Decim v2

Mugi

Multi-S01

Snow 2.0

Asymmetric Cryptographic Scheme

RSASSA-PKCS #1 v1.5

Appendix B

List of cryptographic algorithms that failed Phase 2 evaluation of AKSA MySEAL

Block cipher
SEED

Stream cipher
Enocoro-128 v2
Salsa20/12 (128)
Salsa20/12 (256)
Trivium
ChaCha (128)

Cryptographic Hash Function
Whirlpool
SHA-224
SHA-256
Lesamnta-LW

Digital Signature Scheme
RSA-EMSA2

Deterministic Random Bit Generator
SHA-1 DRBG (160-bit)
SHA-2 DRBG (224-bit)
SHA-2 DRBG (256-bit)
AES-256-CTR-DRBG
Micali-Schnorr-DRBG
2-DEA-MQ-DRBG
3-DEA-MQ-DRBG
AES-MQ-DRBG

Bibliography

[1] R. C. Merkle, Secrecy, authentication, and public key systems. Stanford University, 1979.

[2] K. Aoki, J. Guo, K. Matusiewicz, Y. Sasaki, and L. Wang, "Preimages for Step-Reduced SHA-2," in Advances in Cryptology – ASIACRYPT 2009 (M. Matsui, ed.), vol. 5912 of Lecture Notes in Computer Science, pp. 578–597, Springer-Verlag, 2009.

[3] A. Biryukov, M. Lamberger, F. Mendel, and I. Nikolić, "Second-Order Differential Collisions for Reduced SHA-256," in Advances in Cryptology – ASIACRYPT 2011 (D. H. Lee and X. Wang, eds.), vol. 7073 of Lecture Notes in Computer Science, pp. 270–287, Springer, 2011.

[4] C. Dobraunig, M. Eichlseder, and F. Mendel, "Security Evaluation of SHA-224, SHA-512/224, and SHA-512/256," tech. rep., Graz University of Technology, 2015.

[5] D. Khovratovich, C. Rechberger, and A. Savelieva, "Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family," in Fast Software Encryption, FSE 2012 (A. Canteaut, ed.), vol. 7549 of Lecture Notes in Computer Science, pp. 244–263, Springer, 2012.

[6] J. Li, T. Isobe, and K. Shibutani, "Converting Meet-In-The-Middle Preimage Attack into Pseudo Collision Attack: Application to SHA-2," in Fast Software Encryption, FSE 2012 (A. Canteaut, ed.), vol. 7549 of Lecture Notes in Computer Science, pp. 264–286, Springer, 2012.

[7] F. Mendel, T. Nad, and M. Schläffer, "Improving Local Collisions: New Attacks on Reduced SHA-256," in Advances in Cryptology – EUROCRYPT 2013 (T. Johansson and P. Q. Nguyen, eds.), vol. 7881 of Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 262–278, Springer Berlin Heidelberg, 2013.

[8] H. Yu and D. Bai, "Boomerang Attack on Step-Reduced SHA-512," in Information Security and Cryptology, Inscrypt 2014 (D. Lin, M. Yung, and J. Zhou, eds.), vol. 8957 of Lecture Notes in Computer Science, pp. 329–342, Springer, 2015.

[9] S. S. Keller and L. E. B. III, "The Secure Hash Algorithm 3 Validation System (SHA3VS)," tech. rep., National Institute of Standards and Technology, 2016.

[10] H. Yu, Y. Hao, and D. Bai, "Evaluate the security margins of SHA-512, SHA-256 and DHA-256 against the boomerang attack," Science China Information Sciences, vol. 59, March 2016.

[11] S. Hirose, K. Ideguchi, H. Kuwakado, T. Owada, B. Preneel, and H. Yoshida, "An aes based 256-bit hash function for lightweight applications: Lesamnta-lw," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 95, no. 1, pp. 89–99, 2012.

[12] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "The keccak reference," tech. rep., 2011.

[13] V. Rijmen and E. Oswald, "Update on SHA-1," in Topics in Cryptology - CT-RSA 2005 (A. Menezes, ed.), (Berlin, Heidelberg), Springer Berlin Heidelberg, 2005.

[14] X. Wang, Y. L. Yin, and H. Yu, "Finding Collisions in the Full SHA-1," in Advances in Cryptology – CRYPTO 2005 (V. Shoup, ed.), (Berlin, Heidelberg), pp. 17–36, Springer Berlin Heidelberg, 2005.

[15] "NIST Policy on Hash Functions," 2007. Available at https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions.

[16] "Hash Functions," 2017. Available at `https://csrc.nist.gov/projects/hash-functions/sha-3-project`.

[17] "RFC8017." Available at `https://tools.ietf.org/html/rfc8017`.

[18] "NIST 186-4 Digital Signature Standard." Available at `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf`.

[19] "NIST 186-5 Digital Signature Standard." Available at `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf`.

[20] "ISO/IEC 14888-2:2008 Information technology Security techniques Digital signatures with appendix ." Available at `https://www.iso.org/standard/44227.html`.

[21] "CRYPTREC." Available at `https://www.cryptrec.go.jp/en/list.html`.

[22] "RFC8446." Available at `https://tools.ietf.org/html/rfc8446`.

[23] "Nessie." Available at `https://www.cosic.esat.kuleuven.be/nessie/`.

[24] "RFC3447." Available at `https://tools.ietf.org/html/rfc3447`.

[25] T. Jager, S. A. Kakvi, and A. May, "On the Security of the PKCS#1 v1.5 Signature Scheme," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 18, (New York, NY, USA), p. 11951208, Association for Computing Machinery, 2018.

[26] M. Bellare and P. Rogaway, "The exact security of digital signatures-How to sign with RSA and Rabin," in International conference on the theory and applications of cryptographic techniques, pp. 399–416, Springer, 1996.

[27] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, "Low-Exponent RSA with Related Messages," in Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques, EUROCRYPT96, (Berlin, Heidelberg), p. 19, Springer-Verlag, 1996.

[28] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," IEEE Transactions on Information Theory, vol. 36, no. 3, pp. 553–558, 1990.

[29] D. Boneh, "Twenty Years of Attacks on the RSA Cryptosystem," NOTICES OF THE AMS, vol. 46, pp. 203–213, 1999.

[30] "P1363 homepage," 2000. Available at `https://standards.ieee.org/standard/1363-2000.html`.

[31] A. Standard, "X9. 31-1998," Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), 1998.

[32] F. Liu, C. Dobraunig, F. Mendel, T. Isobe, G. Wang, and Z. Cao, "Efficient Collision Attack Frameworks for RIPEMD-160," in Advances in Cryptology – CRYPTO 2019 (A. Boldyreva and D. Micciancio, eds.), (Cham), pp. 117–149, Springer International Publishing, 2019.