KEMENTERIAN DIGITAL

MySEAL 2.0 Nomination Criteria

[2024]

Name of author: MySEAL FOCUS GROUP

File name: CD-5-RPT-0224-MySEAL 2.0 Nomination Criteria -V1

Date of document: February 2024

Document classification : Public

For inquiry about this document please contact:
Hazlin Abdul Rani
Head, Cryptography Development Department
hazlin@cybersecurity.my

For general inquiry about us or our services,
please email: info@cybersecurity.my

Revision History

DATE	DOCUMENT VERSION	DESCRIPTION OF CHANGES

Preface

This document is issued to provide information to the public regarding the nomination and evaluation criteria for the selection of algorithms to be listed in *Senarai Algoritma Kriptografi Terpercaya Negara* (MySEAL 2.0) / National Trusted Cryptographic Algorithm List. MySEAL 2.0 will be used as a requirement and guideline for the usage of cryptographic algorithms in all trusted cryptography products in Malaysia.

The document is divided into 8 sections. Section 1 introduces the MySEAL 2.0 initiative. Section 2 explains the abbreviations and terms used throughout the document. Sections 3, 4, and 5 cover the general requirements, nomination criteria, and evaluation criteria for each MySEAL primitive, respectively. Section 6 discusses the licensing requirements for cryptographic algorithm nominations. Section 7 outlines the formal nomination requirements, and Section 8 contains additional general information about the MySEAL 2.0 initiative.

This document has been developed by MySEAL Focus Group members, which consists of national cryptographic experts from public universities, private universities, and government agencies. Any inquiries pertaining to this document can be submitted to myseal.fg@cybersecurity.my.

Acknowledgement

Special thanks and appreciation to the following organisations/institutions on the development of this document (in alphabetical order): -

a. Angkatan Tentera Malaysia (ATM)
b. CyberSecurity Malaysia
c. Malaysian Administrative Modernisation and Management Planning Unit (MAMPU)
d. Malaysia Office of the Chief Government Security Officer (CGSO)
e. MIMOS Berhad
f. National Cyber Security Agency (NACSA)
g. Polis Diraja Malaysia (PDRM)
h. Universiti Malaya
i. Universiti Multimedia
j. Universiti Putra Malaysia
k. Universiti Teknikal Malaysia Melaka
l. Universiti Tenaga Nasional
m. Universiti Tunku Abdul Rahman

Table of Contents

1.0 MySEAL Introduction

The information security landscape has undergone significant changes in recent years, with the increasing importance of secure communication and data protection. Recognising the need for trusted cryptographic algorithms, the *Senarai Algoritma Kriptografi Terpercaya Negara* (MySEAL) project was initiated in 2016, laying the foundation for developing and validating cryptographic algorithms in Malaysia. As a result, in 2017, the MySEAL project successfully produced a list of trusted cryptographic algorithms under the AKSA category.

Building upon the achievements of the MySEAL Project, the MySEAL 2.0 initiatives were introduced in 2023. The primary objective remains unchanged: to provide a list of cryptographic algorithms suitable for implementation within the Malaysian context in alignment with the National Cryptography Policy (NCP). While NCP serves as a guiding document for Malaysia to achieve cryptographic sovereignty, MySEAL 2.0 will support the scientific areas of cryptography and cryptanalysis. The continuation of the MySEAL project with the introduction of MySEAL 2.0 initiatives signifies a commitment to evolving technologies and addressing emerging challenges in the field of cryptography.

MySEAL 2.0 initiatives encompass two categories of cryptographic algorithms: AKSA and AKBA. AKSA refers to cryptographic algorithms that have already been published in recognised standards or have undergone thorough evaluation in established cryptography algorithm projects. These algorithms have demonstrated their security, efficiency, and suitability for cryptographic applications. AKSA includes algorithms endorsed by reputable organisations such as FIPS (Federal Information Processing Standards), CRYPTREC (Cryptography Research and Evaluation Committees), NESSIE (New European Schemes for Signatures, Integrity, and Encryption), and The European Network of Excellence in Cryptology (ECRYPT) Stream Cipher Project (eSTREAM), among others.

AKBA refers to new cryptographic algorithms that have not yet been published in recognised standards or widely adopted in the cryptographic community. The AKBA category encourages local cryptographic experts and researchers to explore and develop novel algorithms. By providing a platform for evaluating these new algorithms, MySEAL 2.0 aims to promote the growth and advancement of the Malaysian cryptographic ecosystem. AKBA offers an opportunity for researchers and industry players to contribute cutting-edge algorithms and potentially establish new standards in the field of cryptography.

This document outlines a set of comprehensive criteria that have been established to ensure the integrity and reliability of cryptographic algorithms listed in MySEAL 2.0. These criteria have been carefully developed in accordance with internationally accepted standards and requirements, as defined by the MySEAL Focus Group committee. This committee, led by CyberSecurity Malaysia, comprises esteemed members representing various Malaysian institutions, including academia, government agencies, and industry experts.

MySEAL initiative is by no means a small feat. Ever since the documentation of the National IT Agenda (NITA) in 1996, which listed e-Sovereignty as one of Malaysia's objectives in entering the Information Technology era, the execution of MySEAL has been a significant milestone for Malaysia. It is through this initiative that Malaysia will enter into the realm of information security fundamentals. This challenging arena will attest to Malaysia's perseverance and stamina in protecting its information infrastructure at the cryptographic algorithm level.

Besides providing challenges and aspirations to Malaysian cryptographers, this initiative also aims at nurturing new talent and retaining existing talent. With this note, MySEAL 2.0 initiative has given Malaysia a golden opportunity to provide a collaborative platform between government entities, industries, and higher institutions, to promote and encourage participants in developing new cryptographic algorithms and producing new cryptographers.

The MySEAL initiative is not only a testament to Malaysia's dedication, but it also represents our steadfast pursuit of cryptographic excellence. Through this initiative, we will lay the groundwork for advancing the nation's cryptographic capabilities and ensuring the trustworthiness of our digital ecosystem. By fostering collaboration, research, and innovation, we aim to foster a culture of excellence in developing and utilising cryptographic algorithms.

2.0 Abbreviations and Terms

The following tables describe various abbreviations and terms used throughout this document.

2.1 Abbreviations

No.	Abbreviation	Description
1	AES	Advanced Encryption Standard
2	AKBA	Algoritma Kriptografi Baharu (New Cryptographic Algorithm)
3	AKSA	Algoritma Kriptografi Sedia Ada (Existing Cryptographic Algorithm)
4	API	Application Programming Interface
5	CBC	Cipher Block Chaining
6	CPU	Central Processing Unit
7	DLP	Discrete Logarithm Problem
8	DRBG	Deterministic Random Bit Generator
9	ECB	Electronic Codebook
10	HDK	High Density Key
11	HDP	High Density Plaintext
12	IFP	Integer Factorisation Problem
13	LDK	Low Density Key
14	LDP	Low Density Plaintext
15	MySEAL	Senarai Algoritma Kriptografi Terpercaya Negara (National Trusted Cryptographic Algorithm List)
16	NCP	National Cryptography Policy
17	NIST	National Institute of Standards and Technology
18	PCC	Plaintext / Ciphertext Correlation
19	PKE	Public Key Encryption
20	RAM	Random Access Memory
21	RPRK	Random Plaintext / Random Key
22	S-box	Substitution Box
23	SHA	Secure Hash Algorithm
24	SKA	Strict Key Avalanche
25	SPA	Strict Plaintext Avalanche
26	SIMD	Single Instruction Multiple Data
27	XOR	Exclusive OR

2.2 Terms

No.	Term	Definition
1	AKBA MySEAL	Algoritma Kriptografi Baharu (AKBA) MySEAL, also known as New Cryptographic Algorithms, consists of a list of cryptographic algorithms endorsed by MySEAL, that have not been formally published in recognised standards, cryptographic algorithms listing projects or referenced in the Internet Engineering Task Force (IETF) Standards.
2	AKSA MySEAL	Algoritma Kriptografi Sedia Ada (AKSA) MySEAL, also known as Existing Cryptographic Algorithm MySEAL, consists of a list of cryptographic algorithms endorsed by MySEAL, that have been curated from recognised standards, cryptographic algorithms listing projects and relevant references within the Internet Engineering Task Force (IETF) Standards.
3	Asymmetric Cryptographic	System based on asymmetric cryptographic techniques whose public transformation is used for encryption and whose private transformation is used for decryption. [SOURCE: ISO/IEC 18033-1:2015]
4	Block Cipher	Symmetric encipherment system with the property that the encryption algorithm operates on a block of plaintext, i.e. a string of bits of a defined length, to yield a block of ciphertext. [SOURCE: ISO/IEC 18033-1:2015]
5	Cryptographic Algorithm Listing Projects	Cryptography Research and Evaluation Committees (CRYPTREC), the ECRYPT Stream Cipher Project (eSTREAM) and New European Schemes for Signatures, Integrity and Encryption (NESSIE).
6	Cryptographic Hash Function	Function that maps octet strings of any length to octet strings of fixed length, such that it is computationally infeasible to find correlations between inputs and outputs, and such that given one part of the output, but not the input, it is computationally infeasible to predict any bit of the remaining output. The precise security requirements depend on the application. [SOURCE: ISO/IEC 18033-2:2006]
7	Cryptographic Prime Number Generation	A method for generating and testing prime numbers. Prime numbers are used in various cryptographic algorithms, mainly in asymmetric encryption algorithms and digital signature algorithms. [SOURCE: ISO/IEC 18032:2005]

8	Deterministic Random Bit Generator	Random bit generator that produces a random-appearing sequence of bits by applying a deterministic algorithm to a suitably random initial value called a seed and, possibly, some secondary inputs upon which the security of the random bit generator does not depend. [SOURCE: ISO/IEC 18031:2011]
9	Lightweight Block Cipher	Block ciphers suitable for lightweight cryptography, which are tailored for implementation in constrained environments. NOTE The constraints can be aspects such as chip area, energy consumption, memory size, or communication bandwidth. [SOURCE: ISO/IEC 29192-1:2012] [SOURCE: ISO/IEC 29192-2:2012]
10	Lightweight Cryptographic Hash Function	Lightweight hash function which are tailored for implementation in constrained environments. NOTE The constraints can be aspects such as chip area, energy consumption, memory size, or communication bandwidth. [SOURCE: ISO/IEC 29192-1:2012] [SOURCE: ISO/IEC 29192-5:2012]
11	Must	This word, or the terms “REQUIRED” or “SHALL”, means that the definition is an absolute requirement of the specification. [SOURCE: IETF RFC 2119]
12	MySEAL Focus Group	National cryptographic experts from public universities, private universities, and government agencies in Malaysia that are appointed by CyberSecurity Malaysia.
13	Primitive	A function used to convert between data types. [SOURCE: ISO/IEC 18033-2:2006]
15	Security level	A number associated with the amount of work, such as the number of operations, that is required to break a cryptographic algorithm or system. The security strength is specified in bits and is a specific value from the set {80, 96, 112, 128, 192, 256}. [SOURCE: NIST SP 800-57 Part 1 Revision 5]
16	Should	This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course. [SOURCE: IETF RFC 2119]
17	Standards	Known standards (international or national) such as International Organisation for Standardisation / International

		Electrotechnical Commission (ISO/IEC) and Federal Information Processing Standards (FIPS).
18	Stream Cipher	Symmetric encryption system with the property that the encryption algorithm involves combining a sequence of plaintext symbols with a sequence of keystream symbols one symbol at a time, using an invertible function. [SOURCE: ISO/IEC 18033-1:2015]

3.0 MySEAL 2.0 Requirements

This section presents MySEAL 2.0 requirements which include the categories of cryptographic primitives that are considered in the MySEAL 2.0 initiative, the eligibility of participants, the eligibility of cryptographic algorithms to be submitted and the initiative's general selection criteria required for all primitives. The security analysis, performance efficiency, flexibility, maturity and soundness of justification of cryptographic primitives are further discussed under the sub-section of general selection criteria.

3.1 Categories of Cryptographic Primitives

MySEAL 2.0 is seeking nominations of strong cryptographic primitives in the categories given below:

a) Block Cipher
b) Stream Cipher
c) Asymmetric Digital Signature
d) Asymmetric Encryption
e) Cryptographic Hash Function
f) Cryptographic Prime Number Generation
g) Deterministic Random Bit Generator

3.2 Eligibility of AKSA Nomination

Any cryptographic algorithm that has been published in recognised standards, or published by renowned cryptographic algorithm listing projects or is referenced within relevant Internet Engineering Task Force (IETF) Standards. The MySEAL Focus Group members are responsible for providing the initial list of algorithms to be nominated for AKSA.

The standards organisations and cryptographic algorithm listing projects used as reference in MySEAL are listed below:

a) Standards organisations:
   i. The International Organisation for Standardisation and International Electrotechnical Commission (ISO/IEC)
   ii. The United States Federal Information Processing Standards (FIPS).
b) Cryptographic algorithm listing projects:

i. CRYPTREC
ii. eSTREAM
iii. NESSIE

MySEAL also considers cryptographic algorithms that are referenced within relevant IETF Standards. The algorithms themselves are not necessarily IETF standards but are typically used in IETF Internet standards such as the Transport Layer Security (TLS) and Internet Protocol Security (IPsec). These standards are widely used in practice to provide secure communications.

3.3 Eligibility of AKBA Nomination

Nomination is open to the inventor or owner of a cryptographic algorithm that has not been formally published in recognised standards, cryptographic algorithms listing projects or referenced in the IETF standards.

3.4 MySEAL Selection Process

All cryptographic algorithms nominated for MySEAL undergo a comprehensive evaluation comprising two main phases. In the first phase, the algorithm is assessed based on the nomination criteria. If the algorithm successfully meets the minimum requirements set forth in the nomination criteria, it proceeds to the second phase, wherein a more rigorous assessment is conducted using the following set of evaluation criteria:

a) Security Analysis

This criterion assesses the fundamental cryptographic security level of the algorithm through rigorous analysis.

b) Performance Efficiency

This criterion assesses the algorithm's effective utilisation of computing resources, such as area, time and memory, during hardware and/or software deployment.

c) Flexibility

This criterion assesses the ability of the algorithm to accommodate various parameter sizes, as well as its versatility for implementation across diverse platforms and environments.

d) Maturity

This criterion assesses the algorithm’s acceptance and adoption within the cryptographic and developer communities.

e) Soundness of justification

This criterion assesses the strength and validity of the justifications underlying the design principles of the algorithm.

4.0 Nomination Criteria

This section describes the compliance requirements that should be met by each nominated primitive for both **AKSA** and **AKBA** in Phase 1 of the MySEAL 2.0 selection process. Primitives should satisfy the nomination criteria before undergoing further evaluation in Phase 2 based on the criteria specified in Section 5.0.

4.1 Block Cipher Primitive

Block cipher primitive is divided into two categories; general-purpose block cipher and lightweight block cipher. The nomination criteria are as follows:

4.1.1 General-Purpose Block Cipher

a) Key length of at least 128 bits.

b) Block length of at least 128 bits.

c) Security analysis should include but not limited to:
   - i. Linear cryptanalysis
   - ii. Differential cryptanalysis
   - iii. National Institute of Standards and Technology (NIST) statistical tests

d) Implementation and performance analyses should include, but not limited to, the following metrics for each platform:
   - i. for software platforms: number of processor cycles and program code size.
   - ii. for hardware platforms: throughput, utilisation of FPGA slices and the count of gate equivalents in ASIC.

e) Justification of design principles of the algorithm. See Annex A for an example.

f) Test vectors should include but not limited to:
   - i. Number of keys: - at least 3 for each key size
   - ii. Number of plaintext-ciphertext pairs: - at least 3 for each key size
   - iii. Processing sample must be in Electronic Codebook (ECB) mode with bit ‘0’ padding
   - iv. Intermediate output for each round

4.1.2 Lightweight Block Cipher

a) Key length of at least 80 bits.

b) Block length of at least 64 bits.

c) Security analysis should include but not limited to:
   i. Linear cryptanalysis
   ii. Differential cryptanalysis
   iii. NIST statistical tests

d) Implementation and performance analyses should include, but not limited to, the following metrics for each platform:
   i. for software platforms: number of processor cycles and program code size.
      - Program code size
      - Random Access Memory (RAM) size
   ii. for hardware platforms: throughput, utilisation of FPGA slices and the count of gate equivalents in ASIC.
      - Chip area
      - Cycle
      - Bits per cycle
      - Power
      - Energy

e) Justification of design principles of the algorithm. See Annex B for an example.

f) Test vectors should include but not limited to:
   i. Number of keys: - at least 3 for each key size
   ii. Number of plaintext-ciphertext pairs: - at least 3 for each key size
   iii. Processing sample must be in ECB mode with bit '0' padding
   iv. Intermediate output for each round

4.2 Stream Cipher Primitive

Stream cipher designs can be either software-oriented or hardware-oriented, depending on whether they prioritise flexibility and ease of software implementation or efficient hardware-based encryption performance. The nomination criteria are as follows:

a) For software-oriented stream ciphers:
   i. Key length of at least 128 bits.
   ii. Internal memory of at least 256 bits.

b) For hardware-oriented stream ciphers:
   i. Key length of at least 80 bits.
   ii. Internal memory of at least 160 bits.

c) Security analysis should include but not limited to:
   i. Algebraic attack
   ii. Correlation attack
   iii. Distinguishing attack
   iv. Guess-and-Determine attack
   v. NIST statistical tests

d) Implementation and performance analyses should include, but not limited to, the following metrics for each platform:
   i. for software platforms: number of processor cycles and program code size.
   ii. for hardware platforms: throughput, utilisation of FPGA slices and the count of gate equivalents in ASIC.

e) Justification of design principles of the algorithm. See **Annex C** for an example.

f) Test vectors should include but not limited to:
   i. Number of keys: - at least 3 for each key size
   ii. Number of Initialisation Vectors: - at least 3 for each key size
   iii. Length of keystream: - 256 bits
   iv. Internal state after generating 256 keystream bits

4.3 Asymmetric Digital Signature Primitive

This primitive is divided into Classic Hard Problem-Based Signature Schemes and Stateful Hash-Based Signature Schemes. The nomination criteria are as follows:

4.3.1 Classic Hard Problem-Based Signature Schemes

a) Proof of correctness.

b) Security analysis should include but not limited to:
   i. Hard mathematical problems and assumptions
   ii. Minimum key length needed to achieve the security level[1] of 2^{128}

[1] Security level means the number of steps in the best known attack on a cryptographic primitive

   iii. Security model and its proof[2]

c) Post-Quantum scalability.
d) Efficiency/Complexity analysis should include but not limited to:
   i. In a normal computing environment
   ii. In a constrained environment
e) Justification of design principles of the algorithm.
f) Test vectors should include but not limited to:
   i. Number of key pair: - at least 3 key pairs
   ii. Number of processing samples for each key pair: - at least 2 samples

4.3.2 Stateful Hash-Based Signature Schemes

a) Proof of correctness.
b) Security analysis should include but not limited to:
   i. Hard mathematical problems and assumptions
   ii. Minimum key length needed to achieve the security level of 2^{128}
   iii. Security model and its proof
c) Post-Quantum scalability.
d) Efficiency/Complexity analysis should include but not limited to:
   i. In a normal computing environment
   ii. In a constrained environment
e) Justification of design principles of the algorithm.
f) Test vectors should include but not limited to:
   i. Number of key pair: - at least 3 key pairs
   ii. Number of processing samples for each key pair: - at least 2 samples

4.4 Asymmetric Encryption Primitive

Asymmetric Encryption Primitive is divided into two categories; Encryption Scheme and Key Agreement Scheme. The nomination criteria are as follows:

4.4.1 Encryption Scheme

a) Proof of correctness

[2] Accepted techniques include reduction technique, game-hopping or universal composability

b) Security analysis should include but not limited to:
   i. Hard mathematical problems and assumptions
   ii. Minimum key length needed to achieve the security level of 2^{128}
   iii. Security model and its proof
c) Post-Quantum scalability.
d) Efficiency/Complexity analysis should include but not limited to:
   i. In a normal computing environment
   ii. In a constrained environment
e) Justification of design principles of the algorithm.
f) Test vectors should include but not limited to:
   i. Number of key pair: - at least 3 key pairs
   ii. Number of processing samples for each key pair: - at least 2 samples

4.4.2 Key Agreement Scheme

a) Proof of correctness
b) Security analysis should include but not limited to:
   i. Hard mathematical problems and assumptions
   ii. Minimum key length needed to achieve the security level of 2^{128}
   iii. Security model and its proof
c) Post-Quantum scalability.
d) Efficiency/Complexity analysis should include but not limited to:
   i. In a normal computing environment
   ii. In a constrained environment
e) Justification of design principles of the algorithm.
f) Test vectors should include but not limited to:
   i. Number of key pair: - at least 3 key pairs
   ii. Number of processing samples for each key pair: - at least 2 samples

4.5 Cryptographic Hash Function Primitive

Cryptographic hash function primitive is divided into two categories; general-purpose cryptographic hash function and lightweight cryptographic hash function. The nomination criteria are as follows:

4.5.1 General-purpose Cryptographic Hash Function

a) Digest size of at least 224 bits.

b) Maximum message length 264-1 bits.

c) Security analysis should include but not limited to:

  i. Pre-image resistance
  ii. Second pre-image resistance
  iii. Collision resistance

d) Implementation and performance analyses should include, but not limited to, the following metrics for each platform:

  i. for software platforms: number of processor cycles and program code size.
  ii. for hardware platforms: throughput, utilisation of FPGA slices and the count of gate equivalents in ASIC.

e) Justification of design principles of the algorithm. See Annex D for an example.

f) Test vectors should include but not limited to:

  i. Number of samples for each data size: - at least 3 samples
  ii. Intermediate state value for each round

4.5.2 Lightweight Cryptographic Hash Function

a) Digest size of at least 80 bits.

b) Maximum message length 2^{64}-1 bits.

c) Security analysis should include but not limited to:

  i. Pre-image resistance
  ii. Second pre-image resistance
  iii. Collision resistance

d) Implementation and performance analyses should include, but not limited to, the following metrics for each platform:

  i. for software platforms: number of processor cycles and program code size.
    - Chip area
    - Cycle
    - Bits per cycle
    - Power

- Energy

ii. for hardware platforms: throughput, utilisation of FPGA slices and the count of gate equivalents in ASIC.

- Program code size
- RAM size

e) Justification of design principles of the algorithm. See Annex E for an example.

f) Test vectors should include but not limited to:

i. Number of samples for each data size: - at least 3 samples

ii. Intermediate state value for each round

4.6 Cryptographic Prime Number Generation Primitive

The nomination criteria are as follows:

a) Security analysis should include but not limited to:

i. Probabilistic Prime Generators:

- Prove at least 75% correctness (on par with the Miller Rabin Primality test).
- Primality test should give an output “input is prime”, “input is composite” or “test inconclusive”.
- Run in polynomial time.
- Test vectors should include but not limited to:
    - ➢ Sizes of prime: - Minimum of 512 bits
    - ➢ Number of seeds for each prime size: - 3 seeds (minimum of 128 bits)

ii. Deterministic Prime Generators:

- Proof of correctness
- Run in polynomial time

iii. Able to distinguish Carmichael numbers from prime numbers

iv. Able to generate pseudo primes samples from the generator

v. NIST statistical tests

b) Primality test should include PNT that is listed in MySEAL.

c) Implementation and performance analyses should include, but not limited to, the following metrics for each platform:

i. for software platforms: number of processor cycles and program code size.

ii. for hardware platforms: throughput, utilisation of FPGA slices and the count of gate equivalents in ASIC.

4.7 Deterministic Random Bit Generator (DRBG) Primitive

The nomination criteria are as follows:

a) Security analysis should include but not limited to:

i. DRBG based on asymmetric methodologies:

- Proof of correctness
- Run in polynomial time
- If a DRBG's internal state contains n bits, its period should be at least 2^n.
- Test vectors should include but not limited to:
  - ➢ Sizes of seed: - A minimum of 128 bits
- Utilising strong asymmetric parameters
  - ➢ Integer Factorisation Problem (IFP): - A minimum of 2048 bit
  - ➢ Discrete Logarithm Problem (DLP): - A minimum of 2048 bits

ii. DRBG based on symmetric methodologies:

- Run in polynomial time
- Test vectors should include but not limited to:
  - ➢ Sizes of seed: - A minimum of 128 bits
- Utilising an appropriate symmetric construction

iii. Other types of DRBG:

- Justification of design principles of the algorithm
- Proof of correctness
- Run in polynomial time
- Test vectors should include but not limited to:
  - ➢ Sizes of seed: - A minimum of 128 bits
- NIST statistical tests

iv. Implementation and performance analyses:

- Targeted software platform and/or
- Targeted hardware platform

5.0 Licensing Requirements

This section provides the licensing requirements for nominated algorithms.

1. Nominated algorithms for MySEAL 2.0 should be available without any royalty requirements if selected. In cases where royalty-free availability is not feasible, access to the cryptographic algorithm should be provided in a non-discriminatory manner, without restrictions or biases towards specific users.
2. For AKBA, the nominator must provide a clear statement regarding the intellectual property associated with the nominated algorithm. The intellectual property statement must adhere to the guidelines outlined in Section 7.1 D and should be kept up to date as necessary.

6.0 Formal Nomination Package for AKBA

This section outlines the formal requirements for submitting cryptographic algorithms for nomination in the AKBA MySEAL 2.0 initiative. Submitters and nominated algorithms must comply with the eligibility criteria stated in Section 3.3. This section does not apply to algorithms nominated to AKSA MySEAL as the initial list is provided by the Focus Group as stated in Section 3.2. All cryptographic algorithms nominated for AKBA MySEAL are bound to the following terms and conditions.

a) All submitted cryptographic algorithms, along with any associated information, data, and documents, will be treated as confidential and used solely for the intended purposes outlined in this document.

b) Any external experts engaged for the purposes of evaluation will be bound by strict confidentiality obligations.

c) The MySEAL 2.0 initiative reserves the right to reject submitted cryptographic algorithms that lack clear specification, are not easily comprehensible, or fail to meet the requirements of MySEAL 2.0 in any way.

d) Following the submission of a cryptographic algorithm, the submitter will not receive further communication until the selection process is complete, unless: -

   i. The MySEAL Secretariat requires additional information or supporting documents.

ii. The submitter initiates an enquiry, complaint, or request for clarification that necessitates additional information.

6.1 Algorithm Nomination Package

The following are to be provided with any cryptographic algorithm nomination:

A. Cover sheet with the following information:

[Refer to **Annex G**: Algorithm Nomination Form]

1. Nomination information, either individual or organisation
2. Principal submitter’s name, office telephone number, mobile number, email address and postal address
3. Name(s) of auxiliary submitter(s) (if any)
4. Name of algorithm’s inventor(s)/developer(s)
5. Name of algorithm’s owner (if different from the submitter)
6. Signature of submitter
7. Organisation information (for organisation nomination only)
8. Algorithm’s name
9. Type of submitted algorithm, proposed security level and proposed environment.

B. Algorithm specification and supporting documentation:

1. A complete and unambiguous description of the algorithm in the most suitable form, which may consist of mathematical descriptions, textual diagrams and pseudo-code representations. Specifying the algorithm solely in a programming language is not permitted. Test vectors should be provided in hexadecimal format. For asymmetric algorithms, methods for key generation and parameter selection need to be specified.
2. A statement that there are no hidden weaknesses inserted by the designers. See **Annex G(D.2)**.
3. A statement of the claim on security properties and expected security level, together with an analysis of the algorithm with respect to standard cryptanalytic attacks. Weak keys should also be considered.
4. A statement giving the strengths and limitations of the algorithm.
5. A design rationale that explains design choices.

6. An analysis of the estimated computational efficiency of the algorithms in software and/or hardware platforms.
7. Optionally, information regarding strategies and techniques that can help minimize weaknesses during the implementation of the algorithm.

C. Implementations and test values:

1. A sufficient number of test vectors for each parameter.
2. Reference implementation in C programming language. MySEAL 2.0 will specify a set of cryptographic Application Programming Interface (API) applicable only for symmetric algorithms and hash functions, which will be available at https://mykripto.cybersecurity.my/index.php/services/myseal. The algorithm should implement the API so that the test system can be compatible with all the nominations.
3. Optionally, an optimised implementation for some architecture, a JAVA implementation or an assembly language implementation.

D. Intellectual Property statement:

A statement that gives the position concerning Intellectual Property and the royalty policy for the algorithm (if selected). This statement should include an undertaking to update the MySEAL 2.0 initiative when necessary. See **Annex G(D.3)**.

6.2 Instructions for nomination

1. All nominations should be in either Bahasa Melayu or English and should be supplied in both paper and electronic forms. The electronic form should be in read-only format.
2. The nomination should be sent in two separate media drives. The first media drive contains the Intellectual Property statement and the cover sheet with the following information:

   a) Nomination information, either individual or organisation

   b) Principal submitter's name, office telephone number, mobile number, email address and postal address

   c) Name(s) of auxiliary submitter(s) (if any)

   d) Name of algorithm's inventor(s)/developer(s)

e) Name of algorithm's owner (if different from the submitter)

f) Signature of submitter

g) Organisation information (for organisation nomination only)

h) Algorithm's name

i) Type of submitted algorithm, proposed security level and proposed environment.

3. Second media drive (and any subsequent discs) contains algorithm specification and supporting documentation and implementations and test values.

4. Every media drive must be uniquely labelled. Every media drive must contain a text file labelled "README," listing all files included on the media drive with a brief description of the content of each file. Both paper nomination and media drive should be in one sealed package and labelled as described in **Annex I**.

5. The nomination should arrive at the following address:

   Sekretariat MySEAL
   CyberSecurity Malaysia,
   Level 7, Tower 1,
   Menara Cyber Axis,
   Jalan Impact,
   63000 Cyberjaya,
   Selangor Darul Ehsan,
   Malaysia.

   An acknowledgement will be sent by email within three (3) working days of receipt.

6. Any general questions can be forwarded to myseal.fg@cybersecurity.my. Answers to relevant questions will be posted at https://mykripto.cybersecurity.my/index.php/services/myseal.

7.0 General Information

General information regarding MySEAL 2.0 initiative is available at https://mykripto.cybersecurity.my/index.php/services/myseal.

ANNEX A

Justification on Design Principles of Rijndael

The following statements were extracted from the paper titled **AES Proposal: Rijndael** to give an example statement of design principles. The full paper can be retrieved at http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf.

Design rationale

The three criteria taken into account in the design of Rijndael are the following:

a. Resistance against all known attacks.
b. Speed and code compactness on a wide range of platforms.
c. Design simplicity.

In most ciphers, the round transformation has the Feistel Structure. In this structure typically part of the bits of the intermediate State are simply transposed unchanged to another position. The round transformation of Rijndael does not have the Feistel structure. Instead, the round transformation is composed of three distinct invertible uniform transformations, called layers. By "uniform", we mean that every bit of the State is treated in a similar way.

The specific choices for the different layers are for a large part based on the application of the Wide Trail Strategy, a design method to provide resistance against linear and differential cryptanalysis. In the Wide Trail Strategy, every layer has its own function:

1. **The linear mixing layer**: guarantees high diffusion over multiple rounds.
2. **The non-linear layer**: parallel application of Substitution Boxes (S-boxes) that have optimum worst-case nonlinearity properties.
3. **The key addition layer**: a simple Exclusive OR (XOR) of the Round Key to the intermediate State.

Before the first round, a key addition layer is applied. The motivation for this initial key addition is the following. Any layer after the last key addition in the cipher (or before the first in the context of known-plaintext attacks) can be simply peeled off without knowledge of the key and therefore does not contribute to the security of the cipher. (e.g., the initial and final permutation in the DES). Initial or terminal key addition is applied in several designs, e.g., International Data Encryption Algorithm (IDEA), Secure and Fast Encryption Routine (SAFER) and Blowfish.

In order to make the cipher and its inverse more similar in structure, the linear mixing layer of the last round is different from the mixing layer in the other rounds. It can be shown that this does not improve or reduce the security of the cipher in any way. This is similar to the absence of the swap operation in the last round of the DES.

Motivation for design choices

In the following subsections, we will motivate the choice of the specific transformations and constants. We believe that the cipher structure does not offer enough degrees of freedom to hide a trap door.

The reduction polynomial $m(x)$

The polynomial $m(x)$ (‘$11B$’) for the multiplication in $GF(2^8)$ is the first one of the list of irreducible polynomials of degree 8.

The ByteSub S-box

The design criteria for the S-box are inspired by differential and linear cryptanalysis on the one hand and attacks using algebraic manipulations, such as interpolation attacks, on the other:

1. Invertibility
2. Minimisation of the largest non-trivial correlation between linear combinations of input bits and linear combination of output bits
3. Minimisation of the largest non-trivial value in the XOR table
4. Complexity of its algebraic expression in $GF(2^8)$
5. Simplicity of description

For invertible S-boxes operating on bytes, the maximum input/output correlation can be made as low as 2^{-3} and the maximum value in the XOR table can be as low as 4 (corresponding to a difference propagation probability of 2^{-6}).

By definition, the selected mapping has a very simple algebraic expression. This enables algebraic manipulations that can be used to mount attacks such as interpolation attacks. Therefore, the mapping is modified by composing it with an additional invertible affine transformation. This affine transformation does not affect the properties with respect tot the first three criteria, but if properly chosen, allows the S-box to satisfy the fourth criterion.

We have chosen an affine mapping that has a very simple description per se, but a complicated algebraic expression if combined with the ‘inverse’ mapping. It can be seen as modular polynomial multiplication followed by an addition:

$$b(x) = (x^7 + x^6 + x^2 + x) + a(x)(x^7 + x^6 + x^5 + x^4 + 1) \bmod x^8 + 1$$

The modulus has been chosen as the simplest modulus possible. The multiplication polynomial has been chosen from the set of polynomials coprime to the modulus as the one with the simplest description. The constant has been chosen in such a way that that the S-box has no fixed points ($S - box(a) = a$) and no ’opposite fixed points' ($S - box(a) = \bar{a}$).

Note: other S-boxes can be found that satisfy the criteria above. In the case of suspicion of a trapdoor being built into the cipher, the current S-box might be replaced by another one. The cipher structure and number of rounds as defined even allow the use of an S-box that does not optimise the differential and linear cryptanalysis properties (criteria 2 and 3). Even an S-box that is “average” in this respect is likely to provide enough resistance against differential and linear cryptanalysis.

The MixColumn transformation

MixColumn has been chosen from the space of 4-byte to 4-byte linear transformations according to the following criteria:

1. Invertibility;
2. Linearity in $GF(2)$
3. Relevant diffusion power
4. Speed on 8-bit processors
5. Symmetry
6. Simplicity of description

Criteria 2, 5 and 6 have lead us to the choice to polynomial multiplication modulo $x^4 + 1$. Criteria 1, 3 and 4 impose conditions on the coefficients. Criterion 4 imposes that the coefficients have small values, in order of preference ‘00’, ’01’, ’02’, ’03’…The value ‘00’ implies no processing at all, for ‘01’ no multiplication needs to be executed, ‘02’ can be implemented using *xtime* and ‘03’ can be implemented using *xtime* and an additional XOR. The criterion 3 induces a more complicated conditions on the coefficients.

Branch number

In our design strategy, the following property of the linear transformation of MixColumn is essential. Let F be a linear transformation acting on byte vectors and let the byte weight of a vector be the number of nonzero bytes (not to be confused with the usual significance of Hamming weight, the number of nonzero bits). The byte weight of a vector is denoted by $W(a)$. The Branch Number of a linear transformation is a measure of its diffusion power:

Definition: The branch number of a linear transformation F is

$$min_{a\neq 0}(W(a) + W(F(a))).$$

A non-zero byte is called an active byte. For MixColumn it can be seen that if a state is applied with a single active byte, the output can have at most 4 active bytes, as MixColumn acts on the columns independently. Hence, the upper bound for the branch number is 5. The coefficients have been chosen in such a way that the upper bound is reached. If the branch number is 5, a difference in 1 input (or output) byte propagates to all 4 output (or input) bytes, a 2-byte input (or output) difference to at least 3 output (or input) bytes. Moreover, a linear relation between input and output bits involves bits from at least 5 different bytes from input and output.

The ShiftRow offsets

The choice from all possible combinations has been made based on the following criteria:

1. The four offsets are different and C0 = 0
2. Resistance against attacks using truncated differentials
3. Resistance against the Square attack
4. Simplicity

For certain combinations, attacks using truncated differentials can tackle more rounds (typically only one) than for other combinations. For certain combinations the Square attack can tackle more rounds than others. From the combinations that are best with respect to criteria 2 and 3, the simplest ones have been chosen.

The key expansion

The key expansion specifies the derivation of the Round Keys in terms of the Cipher Key. Its function is to provide resistance against the following types of attack:

- Attacks in which part of the Cipher Key is known to the cryptanalyst.
- Attacks where the Cipher Key is known or can be chosen, e.g., if the cipher is used as the compression function of a hash function.
- Related-key attacks. A necessary condition for resistance against related-key attacks is that there should not be two different Cipher Keys that have a large set of Round Keys in common.

The key expansion also plays an important role in the elimination of symmetry:

- Symmetry in the round transformation: the round transformation treats all bytes of a state in very much the same way. This symmetry can be removed by having round constants in the key schedule.
- Symmetry between the rounds: the round transformation is the same for all rounds. This equality can be removed by having round-dependent round constants in the key schedule.

The key expansion has been chosen according to the following criteria:

- It must use an invertible transformation, i.e., knowledge of any N_k consecutive words of the Expanded Key must allow to regenerate the whole table.
- Speed on a wide range of processors.
- Usage of round constants to eliminate symmetries.
- Diffusion of Cipher Key differences into the Round Keys.
- Knowledge of a part of the Cipher Key or Round Key bits must not allow to calculate many other Round Key bits.
- Enough non-linearity to prohibit the full determination of Round Key differences from Cipher Key differences only.
- Simplicity of description.

In order to be efficient on 8-bit processors, a lightweight, byte-oriented expansion scheme has been adopted. The application of SubByte ensures the non-linearity of the scheme, without adding much space requirements on an 8-bit processor.

Number of rounds

We have determined the number of rounds by looking at the maximum number of rounds for which shortcut attacks have been found and added a considerable security margin. (A shortcut attack is an attack more efficient than exhaustive key search.)

For Rijndael with a block length and key length of 128 bits, no shortcut attacks have been found for reduced versions with more than 6 rounds. We added 4 rounds as a security margin. This is a conservative approach, because:

- Two rounds of Rijndael provide "full diffusion" in the following sense: every state bit depends on all state bits two rounds ago, or, a change in one state bit is likely to affect half of the state bits after two rounds. Adding 4 rounds can be seen as adding a "full diffusion" step at the beginning and at the end of the cipher. The high diffusion of a Rijndael round is thanks to its uniform structure that operates on all state bits. For so-called Feistel ciphers, a round only operates on half of the state bits and full diffusion can at best be obtained after 3 rounds and in practice it typically takes 4 rounds or more.
- Generally, linear cryptanalysis, differential cryptanalysis and truncated differential attacks exploit a propagation trail through n rounds in order to attack $n + 1$ or $n + 2$ rounds. This is also the case for the Square attack that uses a 4-round propagation structure to attack 6 rounds. In this respect, adding 4 rounds actually doubles the number of rounds through which a propagation trail has to be found.

For Rijndael versions with a longer Key, the number of rounds is raised by one for every additional 32 bits in the Cipher Key, for the following reasons:

- One of the main objectives is the absence of shortcut attacks, i.e., attacks that are more efficient than exhaustive key search. As with the key length the workload of exhaustive key search grows, shortcut attacks can afford to be less efficient for longer keys.
- Known-key (partially) and related-key attacks exploit the knowledge of cipher key bits or ability to apply different cipher keys. If the cipher key grows, the range of possibilities available to the cryptanalyst increases.

As no threatening known-key or related-key attacks have been found for Rijndael, even for 6 rounds, this is a conservative margin.

For Rijndael versions with a higher block length, the number of rounds is raised by one for every additional 32 bits in the block length, for the following reasons:

- For a block length above 128 bits, it takes 3 rounds to realise full diffusion, i.e., the diffusion power of a round, relative to the block length, diminishes with the block length.
- The larger block length causes the range of possible patterns that can be applied at the input/output of a sequence of rounds to increase. This added flexibility may allow to extend attacks by one or more rounds.

We have found that extensions of attacks by a single round are even hard to realise for the maximum block length of 256 bits. Therefore, this is a conservative margin.

ANNEX B

Justification on Design Principles of PRESENT

The following statements were extracted from the paper titled **PRESENT: An Ultra-Lightweight Block Cipher** to give an example statement of design principles. The full paper can be retrieved at http://www.ist-ubisecsens.org/publications/present_ches2007.pdf.

Design principles of PRESENT

1. **Goals and environment of use**
   a. The cipher is to be implemented in hardware.
   b. Applications will only require moderate security levels. Consequently, 80 bits security will be adequate. Note that this is also the position taken for hardware profile stream ciphers submitted to eSTREAM.
   c. Applications are unlikely to require the encryption of large amounts of data. Implementations might therefore be optimised for performance or for space without too much practical impact.
   d. In some applications it is possible that the key will be fixed at the time of device manufacture. In such cases there would be no need to re-key a device (which would incidentally rule out a range of key manipulation attacks).
   e. After security, the physical space required for an implementation will be the primary consideration. This is closely followed by peak and average power consumption, with the timing requirements being a third important metric.
   f. In applications that demand the most efficient use of space, the block cipher will often only be implemented as encryption-only. In this way it can be used within challenge-response authentication protocols and, with some careful state management, it could be used for both encryption and decryption of communications to and from the device by using the counter mode.

2. **The permutation layer**
   When choosing the mixing layer, our focus on hardware efficiency demands a linear layer that can be implemented with a minimum number of processing elements, i.e. transistors. This leads us directly to bit permutations. Given our focus on simplicity, we have chosen a regular bit-permutation and this helps to make a clear security analysis.

3. **The S-box**
   We use a single 4-bit to 4-bit S-box S: F42 → F42 in present. This is a direct consequence of our pursuit of hardware efficiency, with the implementation of such an S-box typically being much more compact than that of an 8-bit S-box. Since we use a bit permutation for the linear diffusion layer, AES-like diffusion techniques are not an option for present. Therefore, we place some additional conditions on the S-boxes to improve the so-called avalanche of change.

ANNEX C

Justification on Design Principles of CHACHA20

The following statements were extracted from the paper titled **ChaCha, a variant of Salsa20** to give an example statement of design principles. The full paper can be retrieved at https://cr.yp.to/chacha/chacha-20080120.pdf.

Design principles of ChaCha20

1. Introduction

ChaCha follows the same basic design principles as Salsa20, but I changed some of the details, most importantly to increase the amount of diffusion per round. I speculate that the minimum number of secure rounds for ChaCha is smaller than the minimum number of secure rounds for Salsa20.

This extra diffusion does not come at the expense of extra operations. A ChaCha round has 16 additions and 16 XORs and 16 constant-distance rotations of 32 bits words, just like a Salsa20 round. Furthermore, ChaCha has the same levels of parallelism and vectorizability as Salsa20, and saves one of the 17 registers used by a "natural" Salsa20 implementation. So it is reasonable to guess that a ChaCha round can achieve the same software speed as a Salsa20 round—and even better speed than a Salsa20 round on some platforms. Consequently, if ChaCha has the same minimum number of secure rounds as Salsa20, then ChaCha will provide better overall speed than Salsa20 for the same level of security.

Of course, performance should be measured, not guessed! I wrote and posted new public-domain software for ChaCha, and timed that software, along with the fastest available Salsa20 software, on several computers, using the latest version (20080120) of the eSTREAM benchmarking framework.

2. The quarter-round

ChaCha, like Salsa20, uses 4 additions and 4 XORs and 4 rotations to invertibly update 4 32 bits state words. However, ChaCha applies the operations in a different order, and in particular updates each word twice rather than once. Specifically, ChaCha updates a, b, c, d as follows:

```
a += b; d ^= a; d <<<= 16;
c += d; b ^= c; b <<<= 12;
a += b; d ^= a; d <<<= 8;
c += d; b ^= c; b <<<= 7;
```

3. **The matrix**

ChaCha, like Salsa20/*r*, builds a 4 × 4 matrix, invertibly transforms the matrix through *r* rounds, and adds the result to the original matrix to obtain a 16-word (64-byte) output block. There are three differences in the details. First, ChaCha permutes the order of words in the output block to match the permutation described above. This has no effect on security; it saves time on Single Instruction Multiple Data (SIMD) platforms; it makes no difference in speed on other platforms. Second, ChaCha builds the initial matrix with all attacker-controlled input words at the bottom.

ANNEX D

Justification on Design Principles of Keccak

The following statements were extracted from the paper titled **Keccak sponge function family main document** to give an example statement of design principles. The full paper can be retrieved at https://www.researchgate.net/publication/265099836_Keccak_sponge_function_family_main_document.

1. Choosing the sponge construction

Defining a generic attack:

Definition 1: A shortcut attack on a sponge function is a generic attack if it does not exploit specific properties of the underlying permutation (or transformation).

The Keccak hash function makes use of the sponge construction. This results in the following property:

Provability: It has a proven upper bound for the success probability, and hence also a lower bound for the expected workload, of generic attacks.

The design philosophy underlying Keccak is the hermetic sponge strategy. This consists of using the sponge construction for having provable security against all generic attacks and calling a permutation (or transformation) that should not have structural properties with the exception of a compact description. Additionally, the sponge construction has the following advantages over constructions that make use of a compression function:

a. **Simplicity:** Compared to the other constructions for which upper bounds have been proven for the success of generic attacks, the sponge construction is very simple, and it also provides a bound that can be expressed in a simple way.
b. **Variable-length output:** It can generate outputs of any length and hence a single function can be used for different output lengths.
c. **Flexibility:** Security level can be incremented at the cost of speed by trading in bitrate for capacity, using the same permutation (or transformation).
d. **Functionality:** Thanks to its long outputs and proven security bounds with respect to generic attacks, a sponge function can be used in a straightforward way as a MAC function, stream cipher, a re-seedable pseudorandom bit generator and a mask generating function.

To support arbitrary bit strings as input, the sponge construction requires a padding function. We refer to Section 3.2 of Keccak sponge function family main document for a rationale for the specific padding function we have used.

2. **Choosing an iterated permutation**

   The sponge construction requires an underlying function f, either a transformation or a permutation. f should be such that it does not have properties that can be exploited in shortcut attacks. We have chosen a permutation, constructed as a sequence of almost identical rounds because of the following advantages:

   a. **Block cipher experience:** An iterated permutation is an iterated block cipher with a fixed key. In its design one can build on knowledge obtained from block cipher design and cryptanalysis.
   b. **Memory efficiency:** Often a transformation is built by taking a permutation and adding a feedforward loop. This implies that (at least part of) the input must be kept during the complete computation. This is not the case for a permutation, leading to a relatively small RAM footprint.
   c. **Compactness:** Iteration of a single round leads to a compact specification and potentially compact code and hardware circuits.

1. **Designing the Keccak-f permutations**

   The design criterion for the Keccak-f permutations is to have no properties that can be exploited in a shortcut attack when being used in the sponge construction. It is constructed as an iterated block cipher similar to Noekeon and Rijndael, with the key schedule replaced by some simple round constants. Here we give a rationale for its features:

   a. **Bit-oriented structure Attacks:** Where the bits are grouped (e.g., in bytes), such as integral cryptanalysis and truncated trails or differentials, are unsuitable against the Keccak-f structure.
   b. **Bitwise logical operations and fixed rotations:** Dependence on Central Processing Unit (CPU) word length is only due to rotations, leading to an efficient use of CPU resources on a wide range of processors. Implementation requires no large tables, removing the risk of table-lookup based cache miss attacks. They can be programmed as a fixed sequence of instructions, providing protection against timing attacks.
   c. **Symmetry**: This allows to have very compact code in software and a very compact co-processor suitable for constrained environments.
   d. **Parallelism**: Thanks to its symmetry and the chosen operations, the design is well-suited for ultra-fast hardware implementations and the exploitation of SIMD instructions and pipelining in CPUs.
   e. **Round degree 2:** This makes the analysis with respect to differential and linear cryptanalysis easier, leads to relatively simple (albeit large) systems of algebraic equations and allows the usage of very powerful protection measures against differential power analysis (DPA) both in software and hardware that are not suited for most other nonlinear functions.
   f. **Matryoshka structure:** The analysis of small versions is relevant for larger versions.
   g. **Eggs in another basket:** The choice of operations is very different from that in SHA-1 and the members of the SHA-2 family on the one hand and from AES on the other.

2. Choosing the parameter values

In Keccak, there are basically three security-relevant parameters that can be varied:

a. b: width of Keccak-f,

b. c: capacity, limited by $c < b$,

c. n_r: number of rounds in Keccak-f.

The parameters of the candidate sponge functions have been chosen for the following reasons.

a. $c = 2n$: for the fixed-output-length candidates, we chose a capacity equal to twice the output length n. This is the smallest capacity value such that there are no generic attacks with expected complexity below 2^n.

b. $b = 1600$: The width of the Keccak-f permutation is chosen to favor 64 bits architectures while supporting all required capacity values using the same permutation.

c. Parameters for Keccak[]: for the variable-output-length candidate Keccak[], we chose a rate value that is a power of two and a capacity not smaller than 512 bits and such that their sum equals 1600. This results in $r = 1024$ and $c = 576$. This capacity value precludes generic attacks with expected complexity below 2288. A rate value that is a power of two may be convenient in some applications to have a block size which is a power of two, e.g., for a real-time application to align its data source (assumed to be organised in blocks of size a power of two) to the block size without the need of an extra buffer.

d. $n_r = 24$: The value of n_r has been chosen to have a good safety margin with respect to even the weakest structural distinguishers and still have good performance.

3. The difference between version 1 and version 2 of Keccak

For the 2nd round of the SHA-3 competition, we decided to modify Keccak. There are basically two modifications: the increase of the number of rounds in Keccak-f and the modification of the rate and capacity values in the four fixed-output-length candidates for SHA-3:

a. Increasing the number of rounds of Keccak-f from $12 + l$ to $12 + 2l$ (from 18 to 24 rounds for Keccak-f[1600]): this modification is due to the distinguishers that work on reduced-round variants of Keccak-f[1600] up to 16 rounds. In the logic of the hermetic sponge strategy, we want the underlying permutation to have no structural distinguishers. Sticking to 18 rounds would not contradict this strategy but would leave a security margin of only 2 rounds against a distinguisher of Keccak-f. In addition, we do think that this increase in the number of rounds increases the security margin with respect to distinguishers of the resulting sponge functions and attacks against those sponge functions.

b. For applications where the bitrate does not need to be a power of 2, the new parameters of the fixed-output-length candidates take better advantage of the performance-security trade-offs that the Keccak sponge function allows.

ANNEX E

Justification on Design Principles of SPONGENT

The following statements were extracted from the paper titled **SPONGENT: The Design of Lightweight Cryptographic Hashing** to give an example statement of design principles. The full paper can be retrieved at https://eprint.iacr.org/2011/697.pdf.

The overall design approach for SPONGENT is to target low area while favoring simplicity. The 4-bit S-box is the major block of functional logic in a serial low-area implementation of SPONGENT. It fulfills the present design criteria in terms of differential and linear properties. Moreover, any linear approximation over the S-box involving only single bits both in the input and output masks is unbiased. This aims to restrict the linear hull effect discovered in round-reduced PRESENT.

The function of the bit permutation pLayer is to provide good diffusion, by acting together with the S-box, while having a limited impact on the area requirements. This is its main design goal, while a bit permutation may occupy additional space in silicon. The counters lCounter and ɹǝʇunoƆl are mainly aimed to prevent sliding properties and make prospective cryptanalysis approaches using properties like invariant subspaces more involving.

The structures of the bit permutation and the S-box in SPONGENT make it possible to prove the following differential property:

Theorem 1: Any 5-round differential characteristic of the underlying permutation of SPONGENT with $b \geq 64$ has a minimum of 10 active S-boxes. Moreover, any 6-round differential characteristic of the underlying permutation of SPONGENT with $b \geq 256$ has a minimum of 14 active S-boxes.

The concept of counting active S-boxes is central to the differential cryptanalysis. The minimum number of active S-boxes relates to the maximum differential characteristic probability of the construction. Since in the hash setting there are no random and independent key values added between the rounds, this relation is not exact (in fact that it is even not exact for most practical keyed block ciphers). However, differentially active S-boxes are still the major technique used to evaluate the security of Substitution–Permutation Network (SPN)-based hash functions.

An important property of the SPONGENT S-box is that its maximum differential probability is 2^{-2}. This fact and the assumption of the independency of difference propagation in different rounds yield an upper bound on the differential characteristic probability of 2^{-20} over 5 rounds and of 2^{-28} over 6 rounds for $b \geq 256$ which follows from the claims of Theorem 1.

Theorem 1 is used to determine the number *R* of rounds in permutation π_b: *R* is chosen in a way that π_b provides at least *b* active S-boxes.

ANNEX F

Data Categories for Block Cipher

Ciphertext produced from block ciphers only contains sequence of bits whose length is the block size of the block cipher (e.g ciphertext of LBlock Cipher is 64 bits). However, to evaluate the randomness of cryptographic algorithm, it is important to ensure that ciphertext produced contains a large sequence of bit. To achieve this, Data Categories are used to generate inputs (plaintext or key) for block ciphers to produce ciphertexts that will be concatenated in a certain way. Detail description of the nine categories of data used in block ciphers are provided below:

i. Strict Key Avalanche (SKA)
ii. Strict Plaintext Avalanche (SPA)
iii. Plaintext / Ciphertext Correlation (PCC)
iv. Cipher Block Chaining (CBC) Mode
v. Random Plaintext / Random Key (RPRK)
vi. Low Density Key (LDK)
vii. High Density Key (HDK)
viii. Low Density Plaintext (LDP)
ix. High Density Plaintext (HDP)

1. <u>Strict Key Avalanche (SKA)</u>

The SKA data category is used to examine the sensitivity of block ciphers to changes in the x −bit key. For a fixed plaintext block, avalanche effect is satisfied when any bit of the key is complemented, each bit of the ciphertext block changes with a probability of one half.

Each sample for this data category utilises plaintext that is set to all zero, and X blocks of random x −bit base-keys. The all zero plaintext is first encrypted using each base-key. Next, each base-key is flipped at the i^{th} bit, for $1 \leq i \leq x$ giving a total of $(X * x)$ perturbed-keys. The all-zero plaintext is then encrypted using each perturbed-key. All resultant ciphertexts using pertubed-keys are XORed with the ciphertext resulting from the encryption using its corresponding base-key. Output product of the XOR operation is called a derived block which will be concatenated to construct a large sequence of bits.

2. **Strict Plaintext Avalanche (SPA)**

The SPA data category is used to examine the sensitivity of block ciphers to changes in the $y-$bit plaintext. For a fixed key, avalanche effect is satisfied when any bit of the plaintext is complemented, each bit of the ciphertext block changes with a probability of one half.

Each sample for this data category utilises key that is set to all zero, and Y blocks of random $y-$bit base-plaintexts. Each base-plaintext is first encrypted using the all-zero key. Next, each base-plaintext is flipped at the i^{th} bit, for $1 \leq i \leq y$ giving a total of $(Y * y)$ perturbed-plaintexts. Each perturbed-plaintext is then encrypted using the all-zero key. All resultant ciphertexts of pertubed-plaintexts are XORed with the ciphertext resulting from the encryption of its corresponding base-plaintext. Output product of the XOR operation is called a derived block which will be concatenated to construct a large sequence of bits.

3. **Plaintext / Ciphertext Correlation (PCC)**

The PCC data category is used to examine the correlation between plaintext / ciphertext pairs and is computed using ECB mode of operation.

Each sample for this data category utilises Y blocks of random $y-$bit plaintext and one random $x-$bit key. Each plaintext block is encrypted using the random $x-$bit key. The resultant ciphertext is XORed with its corresponding plaintext. Output product of the XOR operation is called a derived block which will be concatenated to construct a large sequence of bits.

4. **Cipher Block Chaining (CBC) Mode**

The Ciphertext Block Chaining Mode data category is computed using CBC mode of operation. In this data category each block of plaintext is XORed with the previous ciphertext block before being encrypted, whereas the first block is XORed with an initialisation vector. A one-bit change in any plaintext or the initialisation vector will affect all following ciphertext blocks.

Each sample for this data category utilises plaintext that is set to all zero (P), a random $x-$bit key (K), and an all-zeroes initialisation vector (IV). The encryption process is applied for I times. Derived blocks for this data category are ciphertext blocks computed in CBC mode of operation. The first ciphertext block, C_1 is define as $C_1 = E_K(IV \oplus P_1)$, whereas subsequent ciphertext blocks is define as $C_i = E_K(C_{i-1} \oplus P_i)$ for $1 \leq i \leq I$.

5. **Random Plaintext / Random Key (RPRK)**

The RPRK data category is used to examine the randomness of ciphertext based on random plaintext and random key. Each sample for this data category utilises Y blocks of random $y-$bit plaintext and one random x-bit key. Each plaintext block is encrypted using the random $x-$bit key. Derived blocks for this data category are ciphertext blocks computed in ECB mode of operation that will be concatenated to construct a large sequence of bits.

6. **Low Density Key (LDK)**

The LDK data category is formed based on low-density $x-$bit keys. Each sample for this data category utilises Y blocks of random y-bit plaintext and X blocks of specific $x-$bit key. The first plaintext block is encrypted using an all-zeroes $x-$bit key. Then, plaintext blocks are encrypted using x-bit key with a single '1' in each of the x-bit position of the key and all other key bits are set to '0'. This will produce Y_1 blocks of ciphertext. Next, plaintext blocks are encrypted using x-bit key with two '1's in each combination of two bit positions of the key and all other key bits are set to '0'. This will produce C_r^n blocks of ciphertext, where $n = x$ and $r = 2$. In total, derived blocks for this data category are $Y = 1 + Y_1 + C_2^x$ ciphertext blocks computed in ECB mode of operation, and will be concatenated to construct a large sequence of bits.

7. **High Density Key (HDK)**

The HDK data category is formed based on high-density $x-$bit keys. Each sample for this data category utilises Y blocks of random $y-$bit plaintext and X blocks of specific $x-$bit key. The first plaintext block is encrypted using an all-ones $x-$bit key. Then, plaintext blocks are encrypted using $x-$bit key with a single '0' in each of the $x-$bit position of the key and all other key bits are set to '1'. This will produce Y_1 blocks of ciphertext. Next, plaintext blocks are encrypted using $x-$bit key with two '0's in each combination of two bits positions of the key and all other key bits are set to '1'. This will produce C_r^n blocks of ciphertext, where $n = x$ and $r = 2$. In total, derived blocks for this data category are $Y = 1 + Y_1 + C_2^x$ ciphertext blocks computed in ECB mode of operation, and will be concatenated to construct a large sequence of bits.

8. <u>Low Density Plaintext (LDP)</u>

The LDP data category is formed based on low-density y −bit plaintext blocks. Each sample for this data category utilises X blocks of random x −bit keys and Y blocks of specific y −bit plaintext blocks. Firstly, the all-zeroes y −bit plaintext block is encrypted using the first random x −bit key. Then, plaintext blocks with a single '1' in each of the y-bit position of the plaintext and all other plaintext bits are set to '0', is encrypted using other random x −bit keys. This will produce X_1 blocks of ciphertext. Next, plaintext blocks with two ‘1’s in each combination of two bits positions of the plaintext and all other plaintext bits are set to ‘0’, is encrypted using other random x −bit keys. This will produce C_r^n blocks of ciphertext, where $n = y$ and $r = 2$. In total, derived blocks for this data category are $X = 1 + X_1 + C_2^y$ ciphertext blocks computed in ECB mode of operation, and will be concatenated to construct a large sequence of bits.

9. <u>High Density Plaintext (HDP)</u>

The HDP data category is formed based on high-density y −bit plaintext blocks. Each sample for this data category utilises X blocks of random x −bit keys and Y blocks of specific y −bit plaintext blocks. Firstly, the all-zeroes y −bit plaintext block is encrypted using the first random x −bit key. Then, plaintext blocks with a single '0' in each of the y −bit position of the plaintext and all other plaintext bits are set to '1', is encrypted using other random x −bit keys. This will produce X_1 blocks of ciphertext. Next, plaintext blocks with two ‘0’s in each combination of two bits positions of the plaintext and all other plaintext bits are set to ‘1’, is encrypted using other random x −bit keys. This will produce C_r^n blocks of ciphertext, where $n = y$ and r = 2. In total, derived blocks for this data category are $X = 1 + X_1 + C_2^y$ ciphertext blocks computed in ECB mode of operation, and will be concatenated to construct a large sequence of bits.

ANNEX G

BORANG PENCALONAN ALGORITMA ALGORITMA KRIPTOGRAFI BARU (AKBA)

New Cryptographic Algorithm (AKBA) Nomination Form

SENARAI ALGORITMA KRIPTOGRAFI TERPERCAYA NEGARA (MySEAL 2.0)

MAKLUMAT PENCALONAN *Nomination Information*	
☐ Individu *Individual*	☐ Organisasi *Organisation*

A. MAKLUMAT PENCALON

Nominator's Information

MAKLUMAT PENCALON *Nominator's Information*	
NAMA PENCALON UTAMA *Principal Nominator's Name*	
NO TELEFON PEJABAT *Office Tel No*	
NO TELEFON MUDAH ALIH *Mobile No*	
NO FAKSIMILI *Fax No*	
ALAMAT E-MEL *E-mail Address*	
ALAMAT SURAT MENYURAT *Postal Address*	

NAMA PENCALON TAMBAHAN (jika berkenaan) *Name of Auxiliary Nominator(s)* *(if any)*	
NAMA PEREKA CIPTA / PEMBANGUN ALGORITMA *Name of Algorithm Inventor(s) / Developer(s)*	
NAMA PEMILIK ALGORITMA (jika berlainan daripada penyerah utama) *Name of Algorithm's Owner* *(if different from the nominator)*	
TANDATANGAN PENCALON *Signature of Nominator*	

MAKLUMAT ORGANISASI (untuk serahan organisasi sahaja) *Organisation Information (for organisation submission only)*	
ORGANISASI *Organisation*	
ALAMAT *Address*	

B. MAKLUMAT ALGORITMA

Algorithm Information

NAMA ALGORITMA *Name of algorithm*	
PRIMITIF ALGORITMA KRIPTOGRAFI *Cryptographic Algorithm Primitive*	☐ *Block Cipher* ☐ *General-Purpose Block Cipher* ☐ *Lightweight Block Cipher* ☐ *Stream Cipher* ☐ *Synchronous stream cipher* ☐ *Self- Synchronous stream cipher* ☐ *Asymmetric Digital Signature* ☐ *Classic Hard Problem Digital Signature Scheme* ☐ *Stateful Hash-Based Signature Scheme* ☐ *Asymmetric Encryption* ☐ *Encryption Scheme* ☐ *Key Agreement Scheme* ☐ *Cryptographic Hash Function* ☐ *General-Purpose hash function* ☐ *Lightweight hash function* ☐ *Cryptographic Prime Number Generation Primitive* ☐ *Deterministic Random Bit Generator Primitive*
CADANGAN TAHAP KESELAMATAN *Proposed Security Level*	☐ 40 bits ☐ 80 bits ☐ 128 bits ☐ 192 bits ☐ 256 bits ☐ Lain-lain. Sila nyatakan. *Other(s). Please specify.* ______________________
CADANGAN PLATFORM *Proposed Environment*	☐ Perkakasan/*Hardware* ☐ Perisian/*Software*

	☐ *Firmware* ☐ *Hybrid software* ☐ *Hybrid firmware*

C. MAKLUMAT TAMBAHAN

Additional Information

PENGGUNAAN *Implementation*	Sila nyatakan. Sebagai contoh: *Bluetooth, Global System for Mobile communications (GSM), Radio-Frequency Identification (RFID), Smart cards* ____ ____

D. PENYATAAN PENCALON

Statement by the Nominator

1. Nomination statement

☐ I/We do hereby understand that my/our submitted algorithm may not be selected for inclusion in MySEAL 2.0. I/We also understand and agree that after the close of the submission period, my/our submission may not be withdrawn. I/We further understand that I/we will not receive financial compensation from MySEAL 2.0 initiative for my/our nomination.

2. Statement that there are no hidden weaknesses in the algorithm design

☐ I/We certify that, to the best of my knowledge, I/we have fully disclosed there are no hidden weaknesses in my/our algorithm.

☐ I/We hereby enclose information on the known weaknesses of my/our algorithm [……………………………………………… (file/attachment name)]

3. Intellectual Property Statement for the Submission of ……………………………………………… [*name of algorithm*] to the MySEAL 2.0 Initiative

☐ ……………………………………………… [Nominator] currently has patents pending / has not filed for patents on the ……………………………………………… [name of algorithm]. The ……………………………………………… [name of algorithm] is provided royalty-free for commercial and non-commercial use in non-embedded applications. Licenses for use of the ……………………………………………… [name of algorithm] in embedded applications may be obtained from ……………………………………………… [name]. Aside from legal restrictions applying to encryption algorithms (if any), these licenses will be issued on a non-discriminatory basis. We will undertake to update the MySEAL Secretariat when necessary.

Diserahkan oleh (Tandatangan & Cop):
Submitted by (Signature & Stamp):
 	
Tarikh:
Date:
	Diterima oleh (Tandatangan & Cop):
	Received by (Signature & Stamp):
	
	Tarikh:
	Date:

ANNEX H

SENARAI SEMAK
Checklist

Bil *No*	Perkara / Dokumen diperlukan *Document(s) needed*	Disertakan oleh Pencalon (√) *Supplied by Nominator (√)*	Disemak oleh Penerima (√) *Checked by Receiver (√)*	Catatan *Notes*
1	Profil Syarikat *Company Profile*			
2	Laporan analisis *Analysis Report* *a) General-Purpose Block Cipher* ☐ *Linear cryptanalysis* ☐ *Differential cryptanalysis* ☐ Ujian statistik NIST *NIST statistical tests* ☐ Vektor ujian *Test vectors* *(Number of keys: - at least 3 for each key size, number of plaintext-ciphertext pairs: - at least 3 for each key size, processing sample must be in ECB mode with bit '0' padding, and intermediate output for each round).* ☐ Lain-lain. Sila nyatakan. *Other(s). Please specify.* _________________________ *b) Stream Cipher* ☐ *Algebraic attack* ☐ *Correlation attack* ☐ *Distinguishing attack*			

Bil ***No***	**Perkara / Dokumen diperlukan** ***Document(s) needed***	**Disertakan oleh Pencalon (√)** ***Supplied by Nominator (√)***	**Disemak oleh Penerima (√)** ***Checked by Receiver (√)***	**Catatan** ***Notes***
	☐ *Guess-and-Determine attack* ☐ Ujian statistik NIST *NIST statistical tests* ☐ Vektor ujian *Test vectors* *(Number of keys: - at least 3 for each key size, number of Initialisation Vectors: - at least 3 for each key size, length of keystream: - 256 bits, and internal state after generating 256 keystream bits).* ☐ Lain-lain. Sila nyatakan. *Other(s). Please specify.* ________________________ *c) Asymmetric Digital Signature* ☐ *Hard Mathematical Problems and assumptions* ☐ *Security Model and it's proof* ☐ Vektor ujian *Test vectors* *(Number of key pair: - at least 3 key pairs, and number of processing samples for each key pair: - at least 2 samples).* ☐ Lain-*lain*. Sila nyatakan. *Other(s). Please specify.* ________________________ *c) Asymmetric Encryption*			

Bil ***No***	**Perkara / Dokumen diperlukan** ***Document(s) needed***	**Disertakan oleh Pencalon (√)** ***Supplied by Nominator (√)***	**Disemak oleh Penerima (√)** ***Checked by Receiver (√)***	**Catatan** ***Notes***
	☐ *Hard Mathematical Problems and assumptions* ☐ *Security Model and it's proof* ☐ Vektor ujian *Test vectors* *(Number of key pair: - at least 3 key pairs, and number of processing samples for each key pair: - at least 2 samples).* ☐ Lain-*lain*. Sila nyatakan. *Other(s). Please specify.* ________________________ *d) Cryptographic Hash Function* ☐ *Pre-image resistance* ☐ *Second pre-image resistance* ☐ *Collision resistance* ☐ Vektor ujian *Test vectors* *(Number of samples for each data size: - at least 3 samples, and intermediate state value for each round).* ☐ Lain-lain. *Sila* nyatakan. *Other(s). Please specify.* ________________________ *e) Cryptographic Prime Number Generation Primitive* ☐ *Probabilistic Prime Generators* ☐ *Deterministic Prime Generators*			

Bil ***No***	**Perkara / Dokumen diperlukan** ***Document(s) needed***	**Disertakan oleh Pencalon (√)** ***Supplied by Nominator (√)***	**Disemak oleh Penerima (√)** ***Checked by Receiver (√)***	**Catatan** ***Notes***
	☐ *Distinguishing Carmichael numbers from prime numbers* ☐ *Generation of pseudo primes samples from the generator* ☐ Ujian statistik NIST *NIST statistical tests* ☐ Vektor ujian *Test vectors* *(Sizes of prime: - Minimum of 512 bits, and number of seeds for each prime size: - 3 seeds (minimum of 128 bits)).* ☐ Lain-lain. Sila nyatakan. *Other(s). Please specify.* ________________________ *f) Deterministic Random Bit Generator Primitive* ☐ DRBG *based on asymmetric methodologies* ☐ DRBG *based on symmetric methodologies* ☐ DRBG *not based on asymmetric or symmetric methodologies* ☐ Ujian statistik NIST *NIST statistical tests* ☐ *Lain-lain. Sila nyatakan.* *Other(s). Please specify.* ________________________			
3	Laporan prestasi algoritma mengikut keupayaan perkakasan dan/atau perisian *Implementation and performance analyses on hardware and/or software*			

Bil ***No***	**Perkara / Dokumen diperlukan** ***Document(s) needed***	**Disertakan oleh Pencalon (√)** ***Supplied by Nominator (√)***	**Disemak oleh Penerima (√)** ***Checked by Receiver (√)***	**Catatan** ***Notes***
4	Laporan reka bentuk *Justification on design principles*			
5	Vektor ujian *Test vectors*			
6	Penyata / perjanjian / pendedahan Harta Intelek *Intellectual Property statements / agreements / disclosures*			

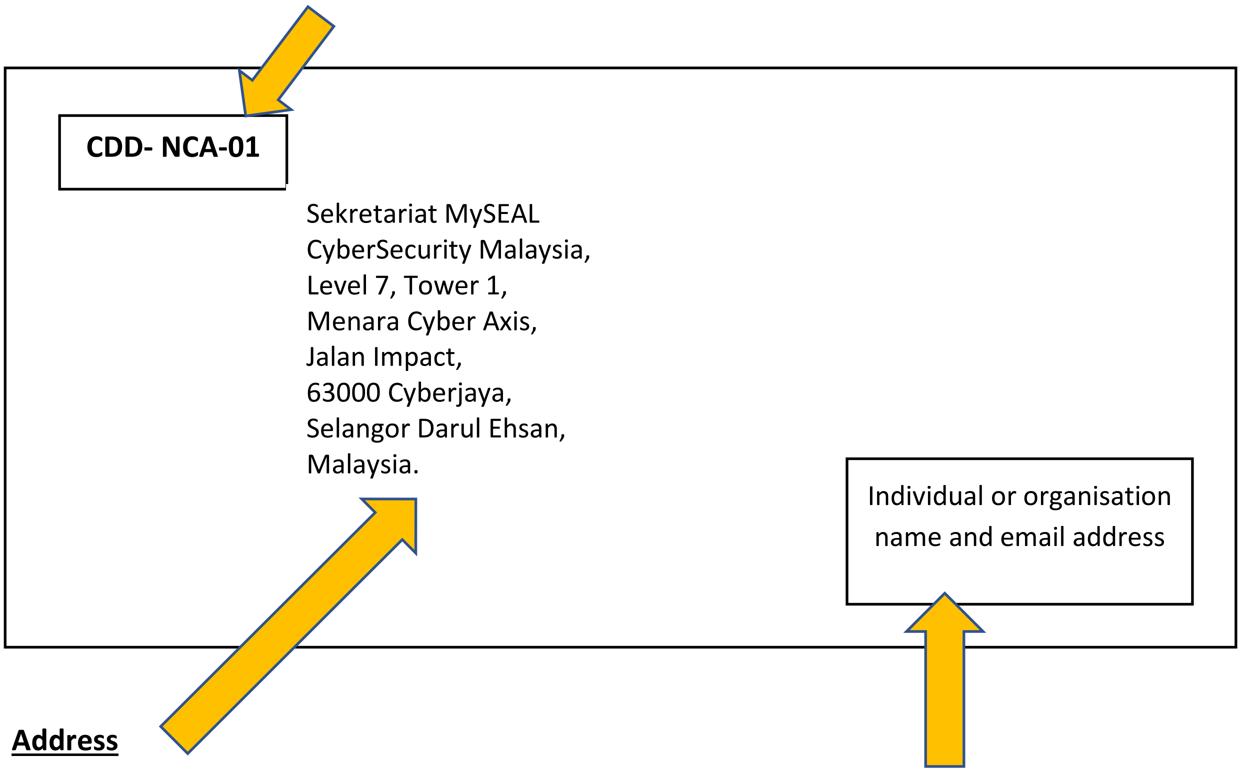
ANNEX I

LABEL GUIDELINE FOR NOMINATION PACKAGE OF MySEAL 2.0 INITIATIVE

PACKAGE FRONT

Nomination Code

Write this code on your package



CDD- NCA-01

Sekretariat MySEAL
CyberSecurity Malaysia,
Level 7, Tower 1,
Menara Cyber Axis,
Jalan Impact,
63000 Cyberjaya,
Selangor Darul Ehsan,
Malaysia.

Individual or organisation name and email address

Address

The official address for nomination

Submitter's Information

Write down the nominator or organisation name and email address