NISTIR 8319

Review of the Advanced Encryption Standard

Nicky Mouha

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8319

National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 8319

Review of the Advanced Encryption Standard

Nicky Mouha
Strativia
Largo, MD

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8319

July 2021

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce for Standards and Technology & Director, National Institute of Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report 8319
21 pages (July 2021)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8319

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Computer Security Division publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: cryptopubreviewboard@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

The field of cryptography continues to advance at a very rapid pace, leading to new insights that may impact the security properties of cryptographic algorithms. The Crypto Publication Review Board ("the Board") has been established to identify publications to be reviewed. This report subjects the first standard to the review process: Federal Information Processing Standard (FIPS) 197, which defines the Advanced Encryption Standard (AES).

Keywords

AES; block cipher; cryptography; FIPS 197; review; Rijndael; standardization.

Acknowledgments

The authors would like to thank Elaine Barker, Lily Chen, Quynh Dang, Morris Dworkin, Jim Foti, John Kelsey, Dustin Moody, Ray Perlner, Andrew Regenscheid, Meltem Sönmez Turan, and Isabel Van Wyk for providing valuable feedback during the development of this publication.

Table of Contents

1 Introduction

The National Institute of Standards and Technology (NIST) develops standards and guidelines for cryptography. In NIST Internal Report (NISTIR) 7977 [42], the development process of these standards and guidelines is laid out. The Crypto Publication Review Board ("the Board") has been established for the periodic review and maintenance of cryptographic standards and guidelines. The project page of the Board is available at: https://csrc.nist.gov/projects/crypto-publication-review-project.

The Advanced Encryption Standard (AES), standardized in FIPS 197 [40], is reviewed in this document. The AES standard is the result of an open competition organized by NIST, where the Rijndael submission by Daemen and Rijmen was selected by NIST as the winner of the competition in 2000 and subsequently standardized as AES in 2001. It is common to refer to both the standard and the algorithm as "AES," and this document will do so as well when the intended meaning is clear from the context.

AES is a block cipher, which is an encryption algorithm that uses a secret key to transform a plaintext into a ciphertext of the same size (referred to as the *block size*). Currently, AES is one of only two block cipher standards that are approved by NIST. The other block cipher standard is the Triple Data Encryption Algorithm (TDEA) [4], commonly known as Triple-DES (Data Encryption Standard). TDEA is now deprecated and will be disallowed after 2023 [6].

Currently, virtually all modern 64-bit processors have native instructions for AES, which includes any recent 64-bit desktop or mobile device. A study by Leech et al. [32] estimated that the economic impact of the development of AES has totaled more than $250 billion over the past 20 years. The use of AES is ubiquitous, and the algorithm enjoys strong support in the cryptographic community.

Examples of protocols and applications that make use of AES are Transport Layer Security (TLS), which is used by virtually all web browsers; Secure Shell (SSH); Internet Protocol Security (IPsec); Wi-Fi; the fourth-generation (4G) Long-Term Evolution (LTE); the fifth-generation New Radio (5G NR); the Zigbee and Bluetooth standards for short-range communications; chip cards (both contact and contactless), including the Europay-MasterCard-Visa (EMV) credit and debit cards; and the Personal Identity Verification (PIV) used across the US Federal Government. For completeness, note that these examples may not use AES in every instance, and that AES may not be supported in older versions of these technologies.

There has been an extensive amount of scrutiny of AES so that the common understanding of the standard now is quite different from when it was first introduced. However, there has not yet been an effort by NIST to consolidate the knowledge gained about AES over the past two decades. This makes AES an ideal candidate to be selected by the Board for the first review.

This review presents an opportunity for retrospection on the development of the standard with an emphasis on new technical insights that are available at the time of the review. It is likely that these technical aspects are much better understood in the years after the finalization of the standard and, therefore, should be evaluated again during the review of the standard.

The core of the review process will be a technical review. However, it is also important to evaluate the standard from an editorial point of view. More specifically, the review process will assess whether the standard is correct, complete, consistent, and unambiguous.

2 Scope

The review of the standard goes well beyond its text. The context in which it is used can be at least as relevant to identify weaknesses, vulnerabilities, or other deficiencies. Therefore, an important challenge will be to define the scope of the review. This can be understood as follows:

- The scope needs to be **broad enough** to capture possible issues with the standard and its uses, identify the implicit and explicit assumptions that are made, and evaluate whether those assumptions are valid for the specific contexts in which the standard is used.
- The scope needs to be **narrow enough** to provide concrete recommendations within a reasonable amount of time.

NIST maintains a variety of resources on cryptographic standards and guidelines and their applications. Examples of the layers of resources for applications of the AES block cipher are given in Table 1.

Table 1: NIST-Maintained Resources for Cryptographic Standards and Guidelines

Type of Resource	Example
Primitive	*AES block cipher [40]*
Higher-level operation	*CMAC (cipher-based message authentication code) mode of operation [21], referencing AES*
Scheme	*Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography [3], referencing CMAC*
Conformance testing (algorithm)	*CMAC Validation System (CMACVS) [28]*
Conformance testing (module)	*CMVP (Cryptographic Module Validation Program) Approved Non-Invasive Attack Mitigation Test Metrics [48]*
Configuration guidelines	*Key length recommendations [6], Transport Layer Security (TLS) recommendations [37], referencing AES*
Vulnerability database	*NIST National Vulnerability Database (NVD) entry on CVE-2014-0160 (Heartbleed), an OpenSSL vulnerability that allows attackers to remotely compromise secret keys [44], such as the AES keys used to encrypt network traffic*

As Table 1 shows, the AES standard is mentioned in several NIST resources. It can be useful to better understand how these NIST resources are organized and how they impact the security properties of implementations of the AES algorithm. In addition to these NIST-maintained resources, there are academic publications and other publicly available resources that describe the security properties of applications of the AES algorithm.

However, only a limited number of attack types will impact the security properties of the AES algorithm itself, so these provide a starting point for the review.

3 The Advanced Encryption Standard

The Advanced Encryption Standard (AES) algorithm [40] transforms an input (e.g., the plaintext) into an output (e.g., the ciphertext) of the same size (referred to as the *block size*). The transformation from plaintext to ciphertext is known as *encryption* and requires the use of a secret key. In the case of AES, the block size is 128 bits, and the secret key can be 128 bits, 192 bits, or 256 bits in length. The inverse transformation from ciphertext to plaintext, known as *decryption*, is defined as well.

The required security properties depend on the application, which is why this document will investigate the NIST standards and guidelines where AES is used and retrospectively examine the AES call for submissions [41] to see whether the properties that were required at the time of submission have been satisfied. To understand the relevance of these security properties, this document will describe whether failure of the security properties leads to attacks.

3.1 Security Requirements

The AES call for submissions [41] stated that one of the factors on which algorithms would be judged is “the extent to which the algorithm output is indistinguishable from a random permutation on the input block.” This means that AES encryption must behave as a “pseudo-random permutation” (PRP). It is often implicit that the inverse operation (decryption) should be indistinguishable from random as well, but this requirement can be made explicit by using the term “strong PRP” or “super PRP.”

Consider, for example, a challenge-response protocol where the challenge is the plaintext, and the response is the ciphertext of the AES operation.[1] An attacker may generate a certain number of challenges (and observe the responses) while having access to the device. If the AES output is distinguishable from random, the attacker would be able to predict the response to a new challenge with a probability higher than 2^{-128} (i.e., higher than the probability of guessing a randomly generated 128-bit ciphertext).

Clearly, the indistinguishability property implies that it should not be possible for the attacker to recover the secret key, which would make it trivial to compute the response to any challenge. This requires an evaluation of whether the AES key lengths (i.e., 128, 192, and 256 bits) are sufficiently long to resist exhaustive key search in the foreseeable future, taking the advent of quantum computing into consideration.

To quantify security in this setting, the number of input-output pairs available to the attacker and the success probability of the attack are relevant as well. Applications may implicitly or explicitly assume that the number of input-output pairs available to the attacker is limited. For example, in the challenge-response protocol mentioned earlier, a randomly generated input will repeat with some probability, and this probability should be sufficiently low for the given

[1] This is a simplified version of a one-way entity authentication protocol, such as the authentication protocol for PIV cards described in Appendix A of NIST Special Publication (SP) 800-73-4 Part 2 [18].

application. Note that, in this example, the number of input-output pairs that is available to the attacker can impact security, even if keys are sufficiently long.

Furthermore, it is important to try to evaluate the “security margin” of AES against various attacks. AES iterates a round function 10, 12, or 14 times, depending on the key size. The security margin indicates the number of “extra” rounds, i.e., the number of rounds beyond those needed to protect AES against the best-known attacks. As explained in [45], the security margin helps to assess the significance of attacks on reduced-round variants of a cryptographic algorithm. Indeed, it is standard practice in cryptanalysis to try to build upon reduced-round attacks. The security of an algorithm cannot be measured by a single figure, however, so the inherent problems in relying on a security margin should be taken into account [45].

Typically, the security margin is evaluated in the single-key setting, i.e., one secret key is drawn uniformly at random. Depending on the application, this may not be appropriate, and other attacks may need to be considered as well. For example, it may be relevant for the application to consider attacks where an attacker can query AES using multiple keys that either have some known relation (related-key attacks) or that are independently drawn at random (multi-key attacks). Multi-key attacks may drastically erode security claims, but a careful analysis is needed, as the mode of operation in which the AES block cipher is used may avoid multi-key degradation [36]; therefore, the impact of multi-key attacks will be discussed in the upcoming review of the NIST-recommended block cipher modes of operation in the 800-38 series of NIST Special Publications.

More generally, the security properties of AES may differ from those of other block ciphers with the same key size and block size, and this may impact current or future applications. For such attacks, bridging the gap between theory and applications may help to decide on the next steps to be taken.

The security of applications not only depends on the theoretical security properties of the cryptographic algorithms but also on the ability to withstand attacks on their implementations. These attacks can either be invasive or non-invasive, depending on whether direct physical contact is required with components inside the device that performs the cryptographic operation.

An attacker may gather key-dependent information through side channels, such as the time taken to perform a computation, or by injecting faults into the computation, such as by using a Rowhammer attack [29] to flip bits in memory without accessing them. When these attacks are performed non-invasively, they can be effective even when there are mechanisms to detect physical tampering of the device.

As the examples in this section show, security not only depends on the algorithmic description of the AES algorithm but also on the way the algorithm is implemented and on how its inputs (plaintexts/ciphertexts, and keys) are generated by the application. When a cryptographic application is vulnerable to a practical attack, this approach allows for the identification of the underlying cause and an evaluation of the vulnerability’s impact.

Table 2: A Selection of Cryptanalysis Results For AES[2]

Key Size	Rounds	Time	Data	Memory	Type	Reference
Any	5	2^{31}	$2^{11.3}$	negligible	yoyo	[47]
	5	$2^{21.5}$	$2^{21.5}$	$2^{21.5}$	partial key recovery	[7]
	12	2^{66}	n/a	2^{64}	known key	[24]
128 bits	7/10	2^{99}	2^{97}	2^{98}	MitM	[20]
	10/10	$2^{126.16}$	2^{88}	2^{8}	biclique	[16]
	10/10	$2^{126.59}$	2	2^{60}	biclique	[15]
	10/10	$2^{125.87}$	2^{72}	2^{60}	biclique	[49]
192 bits	8/12	2^{172}	2^{107}	2^{96}	MitM	[20]
	9/12	$2^{182.5}$	2^{117}	$2^{165.5}$	MitM	[34]
	12/12	2^{176}	2^{123}	2^{152}	related key	[13]
	12/12	$2^{190.16}$	2^{80}	2^{8}	biclique	[16]
	12/12	$2^{190.83}$	2	2^{60}	biclique	[15]
	12/12	$2^{189.76}$	2^{48}	2^{60}	biclique	[49]
256 bits	9/14	2^{203}	2^{120}	2^{203}	MitM	[20]
	10/14	2^{253}	2^{111}	$2^{211.2}$	MitM	[35]
	10/14	2^{45}	2^{44}	2^{33}	related key	[12]
	14/14	2^{131}	2^{131}	2^{65}	related key	[14]
	14/14	$2^{99.5}$	$2^{99.5}$	2^{77}	related key	[13]
	14/14	$2^{254.42}$	2^{40}	2^{8}	biclique	[16]
	14/14	$2^{254.94}$	3	2^{60}	biclique	[15]
	14/14	$2^{254.18}$	2^{40}	2^{60}	biclique	[49]

3.2 Cryptanalysis

A significant number of academic papers have provided analysis of the security of the AES algorithm against cryptographic attacks. These papers typically express the attack complexity in terms of time, data, and memory. Usually, the unit of time is the wall-clock time of a single AES encryption on a classical (i.e., “non-quantum”) computer; the unit of data is a single plaintext-ciphertext pair (encrypted under a secret key) that is obtained by the attacker; and the unit of memory is the size of an AES ciphertext (i.e., a 128-bit block).

A comprehensive study of cryptanalysis attack papers that were published in cryptographic literature was performed. Table 2 contains a list of significant results under various attack

[2] MitM is an abbreviation for “meet-in-the-middle.” Since a known-key attack does not involve a secret key, the data complexity is listed as “n/a” (not applicable). For a discussion of weak-key, known-key, and chosen-key attacks, refer to Grassi et al. [23].

scenarios. This list is based on attack papers and citations in other papers as well as recent papers by Aumasson [1] and Naito et al. [39] that summarize attacks on AES. Focus was placed on identifying what can be considered the most significant attacks rather than compiling an exhaustive list. The attacks are not necessarily comparable; the metrics of the attacks may be computed in different ways, and the attacks may not all have the same success probability. Nevertheless, they give an indication of the current state of the art in AES cryptanalysis.

In Table 2, results that apply for all variants of AES are provided, followed by results that are specific to the variants with 128-bit, 192-bit, and 256-bit key sizes. As this table shows, biclique attacks and attacks outside the single-key setting (e.g., weak-key, known-key, chosen-key, and related-key attacks) exist for the full-round AES. These results are considered cryptographic weaknesses in the sense that they do not apply to all block ciphers with the same key size and block size. The time complexity of the attacks is below exhaustive key search for the given key size.

Biclique attacks perform an exhaustive search over a reduced number of rounds of the cipher and can, therefore, only outperform exhaustive search over all rounds by a small constant factor. It is well known that slight improvements over exhaustive search are always possible (e.g., the "distributive technique" and "early abort technique" [11]); however, biclique attacks provide further speedups that do not apply to every block cipher.

In NIST SP 800-57 Part 1, Revision 5 [2], "security strength" is defined in terms of the number of "operations" to break a cryptographic algorithm. If "operations" can be elementary operations rather than "full-round encryptions," then biclique attacks do not affect the security strength of a cipher, as biclique attacks still perform exhaustive search over a reduced number of rounds.

In a related-key attack, the attacker obtains ciphertexts that are encrypted under multiple keys that are not known to the attacker; however, the attacker can know (or even choose) a mathematical relationship between the keys. Some caution is necessary to avoid related-key attacks that trivially apply to every block cipher; for details, refer to Bellare and Kohno [8] and Mouha and Luykx [38]. For example, the complementation property of DES and Triple-DES [4] enables a straightforward related-key attack: if all of the bits of the plaintext and key are flipped, the ciphertext bits will be flipped with probability 1 instead of probability 2^{-64} for a randomly generated 64-bit ciphertext. In this example, a plaintext-ciphertext pair produced using a secret key allows an attacker to decrypt a ciphertext under a related key. Related-key attacks can be prevented by always generating keys uniformly at random or by imposing additional restrictions when keys are generated with some known relations (e.g., according to a key generation method specified in NIST SP 800-133, Revision 2 [5]).

Meet-in-the-middle attacks proceed by 1) describing the algorithm as a system of mathematical equations and 2) separating the equations into two or more groups such that they can be solved more efficiently than by exhaustive key search. In terms of the number of rounds attacked, meet-in-the-middle attacks seem to outperform all other attacks on AES in the single-key setting, except for biclique attacks. Meet-in-the-middle attacks indicate that AES currently has a security margin of three or four rounds, depending on the key size.

There does not seem to be a consensus in the cryptographic community about what constitutes an acceptable security margin, although a margin of only one round can be considered to be risky. Even if future attacks erode the security margin of AES by an additional round, there would still be a security margin of two or three rounds left, which may be sufficient for the foreseeable future. It seems that the number of rounds for AES was well chosen when it was initially designed and is still widely considered to be more than sufficient by the cryptographic community; a recent paper by Aumasson [1] even suggests reducing the number of rounds.

When Rijndael was selected as the winner of the AES competition, it was stated in [45] that "for 128-bit keys, 6 or 7 out of the 10 rounds of Rijndael have been attacked, the attack on 7 rounds requiring nearly the entire codebook." This statement still holds today in the single-key setting if biclique attacks are excluded: as shown in Table 2, meet-in-the middle attacks reach 7 out of 10 rounds for 128-bit AES with a very high data complexity of 2^{97} plaintext-ciphertext pairs. Therefore, in spite of two decades of public cryptanalysis, the AES algorithm has not been found to be significantly weaker than expected during the standardization process.

Whereas attacking the maximum number of rounds gives an indication of the security margin, there is insight to be gained from focusing on other metrics. These include finding the best attacks for a small number of rounds; attacks with specific bounds on the time, data, or memory requirements; and strong attack settings, such as known-key and chosen-key attacks. The motivation to explore strong attack settings is similar to reduced-round attacks: they give a more fundamental understanding of the security of the AES algorithm and can provide important starting points in the development of other attacks. It is interesting to note that new properties of AES are still being discovered; for example, the new representations of the AES key schedule by Leurent and Pernot [33] had not been described after about 20 years of analysis.

3.3 Parameter Considerations

In the security analysis in the previous section, only cryptanalysis results that outperform generic attacks in terms of their time complexity were considered. Generic attacks are attacks that apply to any block cipher with a given block size and key size.

Determining whether the 128-bit, 192-bit, and 256-bit key sizes of AES provide a sufficient level of security depends on the amount of computing power that will be available at some point in the future. This requires extrapolating the current rate of progress in computing power, which involves uncertainty and risk of error.

Moore's law estimates that the number of transistors on a chip, which roughly corresponds to its computing power, will double every 18 months.[3] This means that in three years, computing power will quadruple, resulting in a loss of two bits of security. If an attacker needs to make a one-time purchase of computing hardware to perform a computation in a timeframe of five years, Moore's law implies that the attacker should wait for three years until the computing power has quadrupled rather than mounting an attack today. Indeed, five years of computation at the current

[3] Scaling down transistors cannot continue indefinitely due to physical limitations. However, the end of Moore's law does not necessarily imply the end of progress in computing. In the absence of information to the contrary, it may be advisable from a security point of view to consider that computing power will continue to increase exponentially in the foreseeable future.

rate is less than two years at a quadrupled rate.

The advent of quantum computing introduces further uncertainty about the amount of computing power that will be available in the future. Due to Grover's algorithm, the number of computations required to perform an exhaustive key search using a quantum computer is roughly the square root of the number of computations required using a classical computer, which corresponds to reducing the security strength by half. In contrast, an attack on the full AES, which would be a cryptanalytical breakthrough, may also reduce the security strength. It is worth noting that multi-key attacks may also reduce the security strength. For an extensive discussion about measuring quantum security, see Section 4.A.5 of the "Call for Proposals" of the NIST Post-Quantum Cryptography project [43].

Parnas's seminal paper on decomposing systems into modules [46] proposes that "one begins with a list of difficult design decisions or design decisions which are likely to change." Modularization plays a central role in software engineering, and it is a principle that can be recognized in NIST's standards and guidelines for cryptography. NIST has reflected this by moving the guidance related to key sizes to SP 800-131A, Revision 2 [6], which specifies the currently acceptable key lengths to be used for AES and other NIST standards.

FIPS 197 provides three variants of AES, each with a different key size: 128, 192, and 256 bits. The 192-bit and 256-bit variants not only have larger key sizes but also require additional iterations of the AES round function — 12 and 14 rounds, respectively — compared to 10 rounds for the 128-bit variant. It is reasonable to expect that the 192-bit and 256-bit variants will always require significantly more computations to attack than the 128-bit variant and may, therefore, provide alternatives in case of significant advances in computing power or cryptanalysis. Currently, however, all three keys sizes are considered secure, and it seems that this will continue to be the case for the foreseeable future.

For all AES variants, the block size is 128 bits. Inputs larger than 128 bits need to be processed using a mode of operation, and the security of the mode of operation is impacted by the block size of the underlying block cipher. The Sweet32 attack [10] shows the practical insecurity of block ciphers with a 64-bit block size and was the basis for the decision to deprecate Triple-DES [4] and disallow it after 2023 [6]. The impact of the 128-bit block size of AES will be analyzed in the upcoming review of the NIST-recommended block cipher modes of operation.

3.4 Implementation Considerations

Depending on the attack setting, there are a wide range of possible implementation attacks. For example, the attack setting could be a desktop or mobile device for which the attacker has full access to the software implementation of AES and the platform on which it is executed [51]. Security in this attack setting is very difficult to achieve, as it is unclear whether there exist software obfuscation techniques that can remain secure for an extended period of time (e.g., to prevent an adversary from extracting the secret key).

The strongest attack setting considered in this document is one in which the AES computation is performed on a device (e.g., a chip card) that may be in the hands of the attacker, but the device can have certain tamper-detection and prevention mechanisms, so the attacker does not have full control of the platform.

Ideally, the calculation of the ciphertext (based on the key and plaintext) is done correctly and without revealing any additional information. In the physical world, however, implementations may inadvertently leak information through side channels or through the injection of faults into the computation.

For example, computations cannot be performed instantaneously but require a certain amount of time. If the execution time depends on the value of the secret key, the implementation may be vulnerable to a timing attack [30]. Timing attacks do not require physical access to the device and may even be performed remotely over the Internet [17]. In addition to execution time, other side channels include power consumption [31] or electromagnetic radiation emanating from the device [22].

When side-channel attacks are performed non-invasively (i.e., without direct physical contact with components within the device), they may evade physical tamper-detection mechanisms. The same applies to fault-injection attacks, such as the Rowhammer attack [29] mentioned earlier. Other fault-injection attacks, such as laser fault injection, may require a partial decapsulation of the integrated circuit (IC) package, which may potentially be detected by tamper-detection mechanisms.

Side-channel attacks were known during the AES standardization process and even explicitly considered as evaluation criteria [45]. It was thought that table lookup operations were not vulnerable to timing attacks, but this statement may not apply to software platforms that use a cache to accelerate the access of data from main memory [9]. It cannot be overstated how devastating side-channel attacks can be on implementations of the AES algorithm compared to cryptanalytic attacks. Attacking AES in the single-key setting requires the equivalent of an exhaustive key search (approximately 2^{128} AES evaluations for the 128-bit key size), which is considered to be infeasible for the foreseeable future. However, a cache-timing attack can recover the secret key of a vulnerable AES implementation in just a matter of minutes [9], making this attack a serious concern if such a side channel is available to the attacker.

It is a nontrivial task to secure an AES implementation against side-channel attacks without sacrificing too much in terms of efficiency. Some notable implementations of AES that are secure against timing attacks are those of Käsper and Schwabe [27] as well as Hamburg [26]. When Intel introduced new instructions for AES, a key motivation was the elimination of timing attacks [25]. Regarding differential-power-analysis (DPA) attacks, the DPA contest [50] is mentioned, in which several DPA-protected implementations of AES were proposed and subjected to public evaluation.

Protection against non-invasive attacks is considered in NIST SP 800-140F [48]. However, SP 800-140F does not yet contain test metrics to protect against non-invasive attacks.

3.5 Editorial Review

FIPS 197 is a highly detailed document that explains the AES algorithm in a clear and unambiguous way, with many additional clarifications to help understand the concepts, including examples, pseudocode, and figures.

Overall, the editorial quality of the document is very high. Nevertheless, there are some potential areas for improvement.

The document does not assume that the reader is familiar with polynomial rings and finite fields and, therefore, explains how the addition and multiplication operations are performed through examples. However, the notation GF(2^8) is used but not explained in FIPS 197. A finite field is also known as a Galois Field (GF), and GF(2^8) refers to a finite field of 256 elements.

FIPS 197 also introduces more mathematical notation than is strictly necessary to understand the mathematical operations. For example, it explains how the **`ShiftRows()`** and **`MixColumns()`** operations can be defined as a multiplication in a polynomial ring over GF(2^8) with the reduction polynomial x^4+1. It is mentioned but not explained that x^4+1 is not an irreducible polynomial over GF(2^8), which might confuse a reader who is not familiar with polynomial rings.

However, the **`ShiftRows()`** operation can also be defined as a byte reordering, and the **`MixColumns()`** operation can be defined as a matrix multiplication (with matrix elements in GF(2^8)). This would have avoided the need to define two symbols for multiplication (using "⊗" and "•"), as only the latter notation would have been needed.

The explanation of the **`AddRoundKey()`** transformation in Section 5.1.4 of FIPS 197 could be improved. That section seems to make a distinction between the "initial Round Key" and the ***Nr*** **`AddRoundKey()`** transformations that follow, where ***Nr*** is the number of rounds of the cipher. However, the pseudocode does not make this distinction and performs a total of ***Nr***+1 **`AddRoundKey()`** transformations.

The key expansion of AES is defined in Section 5.2 of FIPS 197. This section is a bit short and could perhaps benefit from a diagram and some additional explanation. It might be helpful to provide a table with precomputed values of **`Rcon[]`**, similar to the precomputed values of the S-box in Figure 7 and the inverse S-box in Figure 14. However, the values of **`Rcon[]`** do appear in the test vectors in Appendix A.

In Section 5.3.5, an equivalent inverse cipher is defined. It is said that this structure is more efficient than the inverse cipher, but no further details are given to explain the efficiency gains. The efficiency gains in software and hardware could either be briefly explained, or the reader could be referred to the book on AES by Daemen and Rijmen [19].

Lastly, several of the online references in the bibliography are no longer available. However, all information about the AES development process is still available at https://nist.gov/aes, and historic web pages can be found through the Internet Archive (https://archive.org/).

References

[1] Aumasson JP (2020) Too Much Crypto. *Cryptology ePrint Archive preprint,* Report 2019/1492. https://eprint.iacr.org/2019/1492

[2] Barker EB (2020) Recommendation for Key Management: Part 1 – General. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-57 Part 1, Rev. 5. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[3] Barker EB, Chen L, Roginsky AL, Vassilev A, Davis R, Simon S (2019) Recommendation for Pair-Wise Key-Establishment Using Integer Factorization Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-56B, Rev. 2. https://doi.org/10.6028/NIST.SP.800-56Br2

[4] Barker EB, Mouha N (2017) Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-67, Rev. 2. https://doi.org/10.6028/NIST.SP.800-67r2

[5] Barker EB, Roginsky AL, Davis R (2020) Recommendation for Cryptographic Key Generation. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-133, Rev. 2. https://doi.org/10.6028/NIST.SP.800-133r2

[6] Barker EB, Roginsky AL (2019) Transitioning the Use of Cryptographic Algorithms and Key Lengths. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-131A, Rev. 2. https://doi.org/10.6028/NIST.SP.800-131Ar2

[7] Bar-On A, Dunkelman O, Keller N, Ronen E, Shamir A (2018) Improved Key Recovery Attacks on Reduced-Round AES with Practical Data and Memory Complexities. *Advances in Cryptology – CRYPTO 2018*, eds Shacham H, Boldyreva A (Springer, Cham, Switzerland), *Lecture Notes in Computer Science* 10992, pp 185-212. https://doi.org/10.1007/978-3-319-96881-0_7

[8] Bellare M, Kohno T (2003) A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications. *Advances in Cryptology – EUROCRYPT 2003*, ed Biham E (Springer, Berlin), *Lecture Notes in Computer Science* 2656, pp 491-506. https://doi.org/10.1007/3-540-39200-9_31

[9] Bernstein DJ (2005) *Cache-timing Attacks on AES*. Available at https://cr.yp.to/antiforgery/cachetiming-20050414.pdf

[10] Bhargavan K, Leurent G (2016) On the Practical (In-)Security of 64-Bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. *CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (ACM, Vienna, Austria), pp 456-467. https://doi.org/10.1145/2976749.2978423

[11] Biham E, Dunkelman O, Keller N, Shamir A (2011) New Data-Efficient Attacks on Reduced-Round IDEA. *Cryptology ePrint Archive preprint*, Report 2011/417. https://eprint.iacr.org/2011/417

[12] Biryukov A, Dunkelman O, Keller N, Khovratovich D, Shamir A (2010) Key Recovery Attacks of Practical Complexity on AES-256 Variants with up to 10 Rounds. *Advances in Cryptology – EUROCRYPT 2010*, ed Gilbert H (Springer, Berlin), *Lecture Notes in Computer Science* 6110, pp 299-319. https://doi.org/10.1007/978-3-642-13190-5_15

[13] Biryukov A, Khovratovich D (2009) Related-Key Cryptanalysis of the Full AES-192 and AES-256. *Advances in Cryptology – ASIACRYPT 2009*, ed Matsui M (Springer, Berlin), *Lecture Notes in Computer Science* 5912, pp 1-18. https://doi.org/10.1007/978-3-642-10366-7_1

[14] Biryukov A, Khovratovich D, Nikolić I (2009) Distinguisher and Related-Key Attack on the Full AES-256. *Advances in Cryptology - CRYPTO 2009*, ed Halevi S (Springer, Berlin), *Lecture Notes in Computer Science* 5677, pp 231-249. https://doi.org/10.1007/978-3-642-03356-8_14

[15] Bogdanov A, Chang D, Ghosh M, Sanadhya SK (2015) Bicliques with Minimal Data and Time Complexity for AES. *Information Security and Cryptology - ICISC 2014*, eds Lee J, Kim J (Springer, Cham, Switzerland), *Lecture Notes in Computer Science* 8949, pp 160-174. https://doi.org/10.1007/978-3-319-15943-0_10

[16] Bogdanov A, Khovratovich D, Rechberger C (2011) Biclique Cryptanalysis of the Full AES. *Advances in Cryptology – ASIACRYPT 2011*, eds Lee DH, Wang X (Springer, Berlin), *Lecture Notes in Computer Science* 7073, pp 344-371. https://doi.org/10.1007/978-3-642-25385-0_19

[17] Brumley D, Boneh D (2005) Remote Timing Attacks Are Practical. Computer Networks, vol. 48, no. 5, August 2005, pp. 701-16, https://doi.org/10.1016/j.comnet.2005.01.010.

[18] Cooper DA, Ferraiolo H, Mehta KL, Francomacaro S, Chandramouli R, Mohler J (2015) Interfaces for Personal Identity Verification. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-73-4, Includes updates as of February 8, 2016. https://doi.org/10.6028/NIST.SP.800-73-4.

[19] Daemen J, Rijmen V (2002) *The Design of Rijndael* (Springer, Berlin). http://doi.org/10.1007/978-3-662-04722-4

[20] Derbez P, Fouque PA, Jean J (2013) Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. *Advances in Cryptology – EUROCRYPT 2013*, eds Johansson T, Nguyen PQ (Springer, Berlin), *Lecture Notes in Computer Science* 7881, pp 371-387. http://doi.org/10.1007/978-3-642-38348-9_23

[21] Dworkin MJ (2005) Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-38B, Includes updates as of October 6, 2016. https://doi.org/10.6028/NIST.SP.800-38B

[22] Gandolfi K, Mourtel C, Olivier F (2001) Electromagnetic Analysis: Concrete Results. *Cryptographic Hardware and Embedded Systems — CHES 2001*, eds Koç ÇK, Naccache D, Paar C (Springer, Berlin), *Lecture Notes in Computer Science* 2162, pp 251-261. https://doi.org/10.1007/3-540-44709-1_21

[23] Grassi L, Leander G, Rechberger C, Tezcan C (2020) Weak-Key Distinguishers for AES. *Cryptology ePrint Archive preprint*, Report 2019/852 [to appear in the proceedings of SAC 2020]. https://eprint.iacr.org/2019/852

[24] Grassi L, Rechberger C (2017) New and Old Limits for AES Known-Key Distinguishers. *Cryptology ePrint Archive preprint*, Report 2017/255. https://eprint.iacr.org/2017/255

[25] Gueron S (2009) Intel's New AES Instructions for Enhanced Performance and Security. *Fast Software Encryption*, ed Dunkelman O (Springer, Berlin), *Lecture Notes in Computer Science* 5665, pp 51-66. https://doi.org/10.1007/978-3-642-03317-9_4

[26] Hamburg M (2009) Accelerating AES with Vector Permute Instructions. *Cryptographic Hardware and Embedded Systems - CHES 2009*, eds Clavier C, Gaj K (Springer, Berlin), *Lecture Notes in Computer Science* 5747, pp 18-32, https://doi.org/10.1007/978-3-642-04138-9_2

[27] Käsper E, Schwabe P (2009) Faster and Timing-Attack Resistant AES-GCM. *Cryptographic Hardware and Embedded Systems - CHES 2009*, eds Clavier C, Gaj K (Springer, Berlin), *Lecture Notes in Computer Science* 5747, pp 1-17. https://doi.org/10.1007/978-3-642-04138-9_1

[28] Keller SS (2006) *The CMAC Validation System (CMACVS),* Updated August 23, 2011. Available at https://csrc.nist.gov/CSRC/media//Projects/Cryptographic-Algorithm-Validation-Program/documents/mac/CMACVS.pdf

[29] Kim Y, Daly R, Kim J, Fallin C, Lee JH, Lee D, Wilkerson D, Lai K, Mutlu O (2014) Flipping Bits in Memory without Accessing Them: An Experimental Study of DRAM Disturbance Errors. *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)* (IEEE, Minneapolis, United States), pp 361-372. https://doi.org/10.1109/ISCA.2014.6853210

[30] Kocher PC (1996) Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. *Advances in Cryptology - CRYPTO '96*, ed Koblitz N (Springer, Berlin), *Lecture Notes in Computer Science* 1109, pp 104-113. https://doi.org/10.1007/3-540-68697-5_9

[31] Kocher PC, Jaffe J, Jun B (1999) Differential Power Analysis. *Advances in Cryptology — CRYPTO' 99*, ed Wiener M (Springer, Berlin), *Lecture Notes in Computer Science* 1666, pp 388-397. https://doi.org/10.1007/3-540-48405-1_25

[32] Leech DP, Ferris S, Scott JG (2018) The Economic Impacts of the Advanced Encryption Standard, 1996-2017. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Grant/Contract Reports (GCR) 18-017. https://doi.org/10.6028/NIST.GCR.18-017

[33] Leurent G, Pernot C (2020) New Representations of the AES Key Schedule. *Cryptology ePrint Archive preprint*, Report 2020/1253. https://eprint.iacr.org/2020/1253

[34] Li L, Jia K, Wang X (2015) Improved Single-Key Attacks on 9-Round AES-192/256. *Fast Software Encryption*, eds Cid C, Rechberger C (Springer, Berlin), *Lecture Notes in Computer Science* 8540, pp 127-146. https://doi.org/10.1007/978-3-662-46706-0_7

[35] Li R, Jin C (2016) Meet-in-the-Middle Attacks on 10-Round AES-256. *Designs, Codes and Cryptography* 80(3): 459-471. https://doi.org/10.1007/s10623-015-0113-3

[36] Luykx A, Mennink B, Paterson KG (2017) Analyzing Multi-key Security Degradation. *Advances in Cryptology – ASIACRYPT 2017*, eds Takagi T, Peyrin T (Springer, Cham, Switzerland), *Lecture Notes in Computer Science* 10625, pp 575-605. https://doi.org/10.1007/978-3-319-70697-9_20

[37] McKay KA, Cooper DA (2019) Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-52, Rev. 2. https://doi.org/10.6028/NIST.SP.800-52r2

[38] Mouha N, Luykx A (2015) Multi-Key Security: The Even-Mansour Construction Revisited*. Advances in Cryptology — CRYPTO 2015*, eds Gennaro R, Robshaw M (Springer, Berlin), *Lecture Notes in Computer Science* 9215, pp 209-223. https://doi.org/10.1007/978-3-662-47989-6_10

[39] Naito Y, Matsui M, Sakai Y, Suzuki D, Sakiyama K, Sugawara T (2020) *SAEAES,* Submission to the NIST Lightweight Cryptography Project, February 25, 2020. Available at https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/SAEAES-spec-round2.pdf

[40] National Institute of Standards and Technology (2001) Advanced Encryption Standard (AES). (U.S. Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) 197. https://doi.org/10.6028/NIST.FIPS.197.

[41] "Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard," 62 *Federal Register* 48051 (September 12, 1997), pp 48051-48058. https://federalregister.gov/a/97-24214

[42] Cryptographic Technology Group (2016) NIST Cryptographic Standards and Guidelines Development Process. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 7977. https://doi.org/10.6028/NIST.IR.7977

[43] National Institute of Standards and Technology (2016) *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*, December 2016. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[44] National Vulnerability Database (2020) *CVE-2014-0160,* July 2020. Available at https://nvd.nist.gov/vuln/detail/CVE-2014-0160

[45] Nechvatal JR, Barker EB, Bassham LE, Burr WE, Dworkin MJ, Foti JG, Roback E (2001) Report on the Development of the Advanced Encryption Standard (AES). *Journal of Research of the National Institute of Standards and Technology* 106(3), May 2001, pp 511-577. https://doi.org/10.6028/jres.106.023

[46] Parnas DL (1972) On the Criteria to Be Used in Decomposing Systems into Modules. *Communications of the ACM* 15(12): 1053-1058. https://doi.org/10.1145/361598.361623

[47] Rønjom S, Bardeh NG, Helleseth T (2017) Yoyo Tricks with AES. *Advances in Cryptology – ASIACRYPT 2017*, eds Takagi T, Peyrin T (Springer, Cham, Switzerland), *Lecture Notes in Computer Science* 10624, pp 217-243. https://doi.org/10.1007/978-3-319-70694-8_8

[48] Schaffer KB (2020) CMVP Approved Non-Invasive Attack Mitigation Test Metrics: CMVP Validation Authority Updates to ISO/IEC 24759. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-140F. https://doi.org/10.6028/NIST.SP.800-140F

[49] Tao B, Wu H (2015) Improving the Biclique Cryptanalysis of AES. *Information Security and Privacy*, eds Foo E, Stebila D (Springer, Cham, Switzerland), *Lecture Notes in Computer Science* 9144, pp 39-56. https://doi.org/10.1007/978-3-319-19962-7_3

[50] Télécom ParisTech (2020) *DPA contest website.* Available at http://www.dpacontest.org

[51] Wyseur B (2009) White-Box Cryptography. Ph.D. Thesis. (KU Leuven, Leuven, Belgium). Available at https://www.esat.kuleuven.be/cosic/publications/thesis-152.pdf