Algorithms to Support the Evolution of Information Assurance Needs

Introduction

For decades NSA has had the role of ensuring the security of National Security Systems (NSS). NSA's Directorate of Research designs cryptographic algorithms, operating modes, and protocols for this purpose. More recent cultural changes at NSA include an increased reliance on commercial cryptographic technologies for securing NSS and a greater sensitivity to the need to be more transparent and open in our efforts.

In 2013, NSA released a pair of algorithms—Simon and Speck—for the cryptographic community to analyze and review. The goal for Simon and Speck is to provide options that could enable future devices in highly constrained environments to be secured, in particular where current options are not viable, and where classified solutions are not possible.

This look into NSA design efforts is significant and almost unprecedented, and we express our gratitude to the cryptographic and engineering community for their analysis of both the security and performance of Simon and Speck. To provide additional insight and to clarify the appropriate use of Simon and Speck, NSA has decided to publish this document providing some Simon and Speck background and answers to frequently asked questions.

The evolution of Information Assurance

Since its creation in 1952, NSA has maintained the responsibility of securing classified federal telecommunications. Through the decades, this mission has continuously progressed to keep pace with the evolution of technology and the subsequent expansion of the role of information security.

As we consider the future of the NSA IA mission, we see yet another evolution in computing, communications, and networks, which drives new and expanding use of cryptography all around us. We are standing on the precipice of the

Internet of Things (IoT) revolution, an explosion of connected devices that will profoundly affect our daily lives and change the way we interact with the physical world around us. The IoT expands the Internet into a network of smart objects ("things") that have the ability to communicate with each other and with centralized resources. Prominent IoT technologies today include RFID, which is a key enabler of modern supply-chain management and industrial logistics, and wireless sensor networks, whose applications range from home automation to traffic control to medical monitoring.

When we consider the rapid increase in connected devices, it is evident that we are facing unprecedented challenges to security and privacy. For very highly constrained applications, the chosen cryptographic algorithms must be efficient enough to fit within the scarce resources available. Research is currently under way in the growing field of "lightweight" cryptography that strives to protect communication in these environments where limited power, energy, chip area, RAM, ROM, etc. are available, without sacrificing security.

To meet these challenges, NIST has recently announced their intention to create a portfolio of lightweight primitives. Algorithms will be recommended for use only in the context of profiles, which describe physical, performance, and security characteristics. These profiles are intended to capture cryptographic algorithm requirements imposed by devices and applications where lightweight cryptography is needed. NIST will develop these profiles based on community input about application and device requirements for lightweight cryptography.

While this document focuses on the single aspect of encrypting data in constrained devices and networks, it is important to remember that we face many IoT challenges. There is more research and discussion required, in particular to address the unique authentication requirements posed by constrained networks and to consider algorithm agility, especially to manage the potential for unanticipated interoperability needs as networks grow. We welcome the support and involvement of the full breadth of the cryptographic community as we work to address these additional, outstanding challenges.

We aim to further the discussion of lightweight cryptography by proposing Simon and Speck be added to the set of options to consider for particular constrained applications.

SIMON and SPECK

SIMON and SPECK each refer to a separate family of *block ciphers*. A block cipher is an encryption algorithm that allows parties already sharing a common secret key to securely encrypt blocks of data. SIMON and SPECK each support a variety of block and key sizes, so that an algorithm can be tailored to a particular application.

More specifically, each of SIMON and SPECK is a family of *lightweight* block ciphers, meaning they are tailored to be particularly efficient when implemented on extremely constrained platforms, such as RFID tags, sensor nodes, microcontrollers, and other resource-limited devices. It should be noted that while the field of lightweight cryptography includes many primitives, schemes, and protocols specifically designed for efficiency in constrained environments, more general purpose algorithms such as AES may be capable of meeting the needs of many constrained environment applications.

The term "lightweight" does not refer to the security of an algorithm, but only to its suitability for use on highly constrained devices. Thus a secure lightweight block cipher with a given block and key size provides the same level of security as any other secure block cipher with that same block and key size.

A key element of the design of SIMON and SPECK is their flexibility, i.e., their ability to perform well on a full range of hardware and software platforms, and to allow a spectrum of implementations from those with a very small footprint and correspondingly low throughput to those with high throughput and a correspondingly larger footprint. The aim is to support implementations on heterogeneous networks, where good performance may be necessary on many disparate sorts of devices. With that said, SIMON is optimized for hardware applications and SPECK for software applications. So for an application mostly on hardware (respectively software) devices, SIMON (resp. SPECK) is the best choice.

The flexibility of SIMON and SPECK is a by-product of their simplicity. They are built using a set of very simple operations (`AND`, `XOR`, addition, circular shift) that can easily be performed on just about any device capable of computation.

As originally defined, SIMON and SPECK offered a broad range of block and key sizes:

block size	key size
32	64
48	72, 96
64	96, 128
96	96, 144
128	128, 192, 256

When we consider potential NSS applications, SIMON 128/256 and SPECK 128/256, which act on 128-bit plaintext blocks using a 256-bit key, are the only suitable options.

Full descriptions of the algorithms can be found at [BSS+13].

SIMON AND SPECK FAQ

Here follows an FAQ for the algorithms SIMON and SPECK.

1. GENERAL QUESTIONS

How do SIMON and SPECK factor into NSA's plans for constrained applications?

NSA's Directorate of Research designs cryptographic algorithms, operating modes, and protocols in support of NSA's Information Assurance Mission to ensure the security of National Security Systems (NSS). It is likely that future NSS will include constrained, commercial devices. The goal of SIMON and SPECK is to provide options that could enable future devices to be secured, in particular where current options are not viable, and where classified solutions are not possible.

Many lightweight block ciphers have been proposed, but often these block ciphers target a particular platform (often an ASIC). The aim of SIMON and SPECK is to provide strong performance on multiple platforms, including future, *unforeseen* platforms.

NSA has offered SIMON and SPECK to the cryptographic community in the hope that they will prove useful to the development of a robust set of algorithms

suited to the full array of future constrained applications. We look to NIST and input from NSS operators to help define the appropriate role each algorithm should play in this space.

When should Simon and/or Speck be used?

AES is the standard for National Security Systems. For most usages, lightweight cryptography is unnecessary, and one should implement AES-256.

Interoperability is an important consideration, and is another reason to prefer AES-256.

For some highly constrained applications, Simon or Speck may be the best option. This can especially be the case for a closed system where interoperability is not an issue or for systems where interoperability is achieved by communication with a more powerful device (which would be able to encrypt and decrypt using any number of supported algorithms, e.g. Simon, Speck, or AES).

Defining the use cases where specifically tailored lightweight algorithms will be needed is a current topic of interest. A discussion of military IoT applications, where lightweight cryptography may be appropriate, can be found here in Leveraging the Internet of Things for a More Efficient and Effective Military from the Center for Strategic and International Studies (CSIS). NIST is considering applications of lightweight cryptography for sensor networks, healthcare, the smart grid, etc.; see NIST's Lightweight Cryptography Project. NASA has expanding programs for extremely small satellites such as CubeSats which may have need for lightweight algorithms; see NASA's definition of SmallSats and CubeSats.

2. Usage issues

Are Simon and Speck interoperable?

No. Different block ciphers do not interoperate. Thus data encrypted with Speck 128/256, for example, must be decrypted with Speck 128/256. Similarly, data encrypted with AES-256 must be decrypted with AES-256.

How would Simon and Speck integrate into secure systems?

When considering a new system design, it is important to remember that many factors come into play. SIMON and SPECK are cryptographic primitives that are meant to be used as components of highly constrained security systems. There is much in addition to the block cipher that is necessary to create a secure system. Key management is fundamental to secure systems, but is separate from the choice of block cipher used.

Similarly, authentication remains an important issue for constrained systems as traditional solutions may impose undue burdens. Protocols to define the full functioning of the system will be required; for some applications there may be existing profiles for utilizing AES, but new protocols may need to be developed for highly constrained environments.

How keys are generated, distributed, revoked, updated, etc. is of critical importance, but is also very much dependent on the particular system, so it's not possible to provide a general solution. See NIST Special Publication 800-130 for key management recommendations. Additional work will be required to define lightweight authentication mechanisms, protocols, and system profiles.

Are SIMON and SPECK standardized?

SIMON and SPECK are going through the standardization process with the International Organization for Standardization (ISO). ISO's lightweight cryptography standard (ISO/IEC 29192-2) currently supports PRESENT (which has a 64-bit block and allows 80- or 128-bit keys) and CLEFIA (which has a 128-bit block and allows 128-, 192, or 256-bit keys). In the Fall of 2014 an amendment was proposed to include SIMON and SPECK in this standard.

We note that the ISO/IEC 29192-2 requires a minimum key size of 80 bits, and so the two smallest sizes of SIMON and SPECK are not included in the proposed amendment. The smallest key size supported by the SIMON and SPECK variants proposed for the ISO amendment is 96 bits. Moreover, because of concerns expressed about small block sizes, there are active discussions regarding whether or not to include the 48-bit block versions.

In addition, SIMON and SPECK have been proposed for ISO's RFID air interface standard, ISO/IEC 29167.

NIST has recently announced their intention to create a portfolio of lightweight primitives through an open process similar to the selection of modes of operation of block ciphers; see NISTIR 8114. We plan to submit appropriate variants of Simon and Speck for consideration in the NIST process. As the minimum key size required by NIST is 112 bits, only those variants with key size at least 128 bits are expected to be appropriate for the NIST lightweight primitive selection process.

Are Simon and Speck suitable for use in NSS?

Simon and Speck have been deemed to provide the security necessary for National Security Systems. NSS operators and vendors who do not believe that they can support AES in their systems and products should contact the IAD Client Contact Center (phone 410-854-4200, email `IAD_CCC@nsa.gov`); we will work with you to examine your use cases and explore the possible use of Simon or Speck in your systems.

At the time of writing, Simon and Speck are not included in the Commercial National Security Algorithms (CNSA) Suite, and a decision has not been made regarding their future inclusion. Until such a time, any NSS usage would need to be coordinated with the IAD Client Contact Center. It should also be noted that only Simon 128/256 and Speck 128/256 achieve the security strength required for NSS.

Is a license required to use Simon or Speck?

The algorithms are free for anyone to use. There are no patent or licensing restrictions. Copyright and related rights are expressly waived through the CC0 1.0 Universal License.

How do I verify my implementation of Simon and/or Speck?

The specification paper [BSS+13] provides test vectors for each of the versions of Simon and Speck. Reference C code, Cryptol code, fast x86 code, and other information can be found at

`https://github.com/iadgov/simon-speck`

3. Technical issues

Are Simon and Speck secure?

We have a high level of confidence in the security of Simon and Speck. There has been a large amount of cryptanalysis done to date on these algorithms. No vulnerabilities have been found.

We expand on this answer below, where we address two subsections. The first concerns the security of the *algorithms*, which is a mathematical question. The second concerns security of *implementations* of the algorithms, which is an engineering question.

Security of the algorithms. For a block cipher to be secure there should in particular be no *key-recovery attack*,* i.e., no more efficient way to recover the secret key than to exhaustively try all possible keys. The *expected* work to find the correct key is approximately equal to the (optimized) work to do 2^{k-1} encryptions, where k is the key size in bits, and any method of recovering the secret key with less work—even given access to all the inputs and outputs to the block cipher—is considered an attack.

Consensus regarding the security of a block cipher emerges when the algorithm has been thoroughly studied by experts in the field. To facilitate this, it is important that the full details of the algorithm be made available to the cryptographic community. Simon and Speck were published to the IACR ePrint archive (paper #2013/404) in June 2013 to allow leading experts around the world an opportunity to analyze these algorithms.

Simon and Speck are extremely simple and easy to understand designs, and this makes them attractive cryptanalytic targets. As of August 2016, more than 50 cryptanalysis papers and theses on these algorithms have been published, and no weaknesses have been found.

Much of the cryptanalysis of Simon and Speck has been carried out by the world's leading academic cryptographers. Some notable papers include [Din14], [WWJZ14], [CW15], [SHY16], [TM16], [FCC+16], and [BVC16]. We thank the

*We note that there are further security notions, and that a full discussion of block cipher security is beyond the scope of this note. We refer the reader to the academic literature for more on this topic.

researchers who have carried out all of this excellent work, and we encourage further research on SIMON and SPECK, as additional independent analysis will serve to further raise confidence in the security of the algorithms.

Security of implementations. Side channel attacks (whereby secret information is recovered by measuring electromagnetic emanations, timing, power, etc.) can be of serious practical concern. Such attacks are attacks on *implementations* rather than on algorithms per se, and usually require close physical access to a device. Straightforward implementations of cryptographic algorithms typically will leak information via side channels. For a particular application it can be important to use implementations that provide side channel protection, but such implementations tend to drive up area, power usage, etc.

Because of their simplicity, SIMON and SPECK have been shown by a number of researchers to offer relatively simple side channel mitigations. See, for example, [STE15], [CITE15], and [BGDN14].

SIMON and SPECK are naturally immune to *cache timing attacks*, which can be an issue for algorithms which use Sboxes. They have also been shown in [BDG16] to provide significantly better resistance to *correlation power analysis* on small microcontrollers than Sbox-based lightweight block ciphers.

Why would I choose one over the other?

If AES-256 can be implemented, it should be.

In cases where an application is so constrained as to render the implementation of AES-256 infeasible, SIMON or SPECK may be a good alternative. Both offer strong performance on a variety of platforms. However, for software platforms (microcontrollers and desktop processors), SPECK outperforms SIMON. On hardware platforms (ASICs and FPGAs), SIMON outperforms SPECK. Therefore, when using lightweight cryptography in a constrained mostly software-based environment, SPECK is the natural choice. For applications mostly on constrained hardware, SIMON is the natural choice. For constrained heterogeneous networks, with mixed hardware and software devices, either works well.

Performance data can be found in [BSS+15] and [BSS+14]. Microcontroller performance data for various lightweight block ciphers has been compiled by

researchers at the University of Luxembourg as part of their *Fair Evaluation of Lightweight Cryptographic Systems* (FELICS). These results can be found at

`https://www.cryptolux.org/index.php/FELICS_Block_Ciphers`.

What mode should I use?

A block cipher is used in a mode; this describes how blocks are processed into ciphertext using the block cipher. The most basic mode is *electronic codebook mode*, whereby the data to be encrypted is parsed into pieces of size equal to the block size of the cipher, each of which is encrypted and transmitted as ciphertext. This straightforward mode is often not appropriate, because repeated plaintext blocks produce repeats in ciphertext blocks, and this reveals some information about the structure of the plaintext.

The mode one should use depends on a variety of factors. For a discussion and recommendations, see NIST Special Publication 800-38A.

A further consideration is whether data *authentication* is required. Used in standard modes, a block cipher provides *confidentiality*, but supplies no cryptographic means for detecting whether blocks of ciphertext were corrupted in transit. If this functionality is required, then an authenticated encryption mode should be used. (And it should be noted that there is overhead associated with using authenticated encryption modes.)

4. More information

Where can I get more information?

The designers of Simon and Speck have published several papers. The algorithms are specified in [BSS+13]. Additional performance data and some of the design rationale is given in [BSS+15]. Performance on AVR 8-bit microcontrollers is discussed in [BSS+14].

Many papers by other authors on the cryptanalysis and implementation of Simon and Speck can be found on the International Association for Cryptologic Research (IACR) Cryptology ePrint archive: `https://eprint.iacr.org`.

A selected bibliography is included in this document. Additional references can be found by doing a Google Scholar search.

For additional questions, including questions related to the use of SIMON and SPECK, please contact the

IAD Client Contact Center
phone: 410-854-4200
email: IAD_CCC@nsa.gov

REFERENCES

[BDG16] Alex Biryukov, Daniel Dinu, and Johann Großschädl. *Correlation Power Analysis of Lightweight Block Ciphers: From Theory to Practice*, pages 537–557. Springer, 2016.

[BGDN14] Shivam Bhasin, Tarik Graba, Jean-Luc Danger, and Zakaria Najm. A Look into SIMON from a Side-Channel Perspective. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2014*, pages 56–59, May 2014.

[BSS+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. http://eprint.iacr.org.

[BSS+14] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Block Ciphers on AVR 8-bit Microcontrollers. Cryptology ePrint Archive, Report 2014/947, 2014. http://eprint.iacr.org.

[BSS+15] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. SIMON and SPECK: Block Ciphers for the Internet of Things. Cryptology ePrint Archive, Report 2015/585, 2015. http://eprint.iacr.org.

[BVC16] Alex Biryukov, Vesselin Velichkov, and Yann Le Corre. Automatic Search for the Best Trails in ARX: Application to Block Cipher Speck. In *Fast Software Encryption (FSE 2016 Proceedings)*, Lecture Notes in Computer Science. Springer-Verlag, 2016.

[CITE15] Cong Chen, Mehmet Sinan Inci, Mostfa Taha, and Thomas Eisenbarth. SpecTre: A Tiny Side-Channel Resistant Speck Core for FPGAs. Cryptology ePrint Archive, Report 2015/691, 2015. http://eprint.iacr.org.

[CW15] Huaifeng Chen and Xiaoyun Wang. Improved Linear Hull Attack on Round-Reduced SIMON with Dynamic Key-guessing Techniques. Cryptology ePrint Archive, Report 2015/666, 2015. http://eprint.iacr.org.

[Din14] Itai Dinur. Improved Differential Cryptanalysis of Round-Reduced Speck. Cryptology ePrint Archive, Report 2014/320, 2014. http://eprint.iacr.org.

[FCC+16] Kai Fu, Tingting Cui, Huaifeng Chen, Ling Sun, Long Wen, and Andrey Bogdanov. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In *Fast Software Encryption (FSE 2016 Proceedings)*, Lecture Notes in Computer Science. Springer-Verlag, 2016.

[SHY16] Ling Song, Zhangjie Huang, and Qianqian Yang. Automatic Differential Analysis of ARX Block Ciphers with Applications to SPECK and LEA. Cryptology ePrint Archive, Report 2016/209, 2016. http://eprint.iacr.org.

[STE15] Aria Shahverdi, Mostfa Taha, and Thomas Eisenbarth. Silent Simon: A Threshold Implementation under 100 Slices. Cryptology ePrint Archive, Report 2015/172, 2015. http://eprint.iacr.org.

[TM16] Yosuke Todo and Masakatu Morii. Bit-Based Division Property and Application to Simon Family. Cryptology ePrint Archive, Report 2016/285, 2016. http://eprint.iacr.org.

[WWJZ14] Ning Wang, Xiaoyun Wang, Keting Jia, and Jingyuan Zhao. Differential Attacks on Reduced SIMON Versions with Dynamic Key-guessing Techniques. Cryptology ePrint Archive, Report 2014/448, 2014. http://eprint.iacr.org.