# E3 Protocol White Paper

## Security Analysis of PKMS2 Explained

12/15/17

# 1 Introduction

Encryption is a critical element of the global datascape. It offers privacy and security protections, is critical to building and maintaining trust, provides a sense of stability and solidity, empowers commerce, and is the foundation of ubiquitous connectivity. Within the ever-expanding ocean of data being created by this connectivity is a similarly growing sea of sensitive information that includes anything from healthcare records to financial-account information, critical business-planning materials, intellectual property, trade secrets, contracts, and contact lists.

Concomitantly increasing are the number of attacks. Gemalto's Breach Level Index[1] indicates a steady year-over-year growth in breaches. Perhaps the most interesting statistic provided by the Breach Level Index is that only 4% of the 9.2 billion records stolen since 2013 were encrypted (and therefore useless to the data thieves). The good news is that both enterprises and individuals are employing encryption on a more frequent basis to protect their sensitive information. The bad news is that encryption technology itself is under attack. Advances in cryptanalytic research[2] and quantum computing threaten the security guarantees provided by existing encryption mechanisms.[3]

The Secure Channels Pattern Key, Multi-Segment, Multi-Standard (PKMS2) encryption protocol is a patented mode of operation for use with symmetric block ciphers that provides improved security guarantees over conventional ciphers such as the Advanced Encryption Standard (AES)[4] or ARIA.[5] Specifically, PKMS2 provides improvements in three areas:

- It relies on multiple ciphers, and provides "fallback security" in case one or more of those ciphers is broken by some future attack;
- It provides an effective key length that is significantly longer than conventional ciphers
- Through the use of segmentation, it substantially increases the effort required to attack the encryption scheme and recover the underlying data.

A mathematical analysis of PKMS2 was conducted by our Vice President of Cryptographic Engineering and verified by three prominent academic cryptographers. This paper is a plain-English companion to those reports, and explains what PKMS2 is and what the analysis implies about its security guarantees.

# 2 Data Confidentiality and Encryption

Modern cryptography addresses three fundamental security properties: Confidentiality, Integrity, and Authenticity/Non-Repudiation.

Confidentiality is about protecting information from unauthorized disclosure. Information is valuable: bank account statements, credit card numbers, medical histories, trade secrets, intellectual property, or military plans all have intrinsic value to both legitimate and non-legitimate actors. Ensuring that the information is protected is often more than just prudent, it may be a legal requirement. A number of laws require confidentiality, including the Health Insurance Portability and Accountability Act (HIPAA),

[1] http://breachlevelindex.com/
[2] For an example, see: https://www.hindawi.com/journals/jece/2017/9828967/abs/
[3] http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf
[4] http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
[5] https://tools.ietf.org/html/rfc5794

the Health Information Technology for Economic and Clinical Health (HITECH) Act, the Privacy Act of 1974, Section 208 of the E-Government Act of 2002, the Family Educational Rights and Privacy Act (FERPA), the Children's Online Privacy Protection Act (COPPA) and the Gramm-Leach-Bliley Act. Failures to ensure data confidentiality can lead to reputational damage, legal action, and in some cases result in an enterprise's closure.[6]

Encryption ensures confidentiality by hiding information about a message from eavesdroppers or other unauthorized parties. Cryptography's goal is not to hide the existence of a message but to hide its meaning.[7] An encrypted message is one that has been scrambled according to a predetermined algorithm that has been agreed upon by the sender and recipient, using a secret *key* they have shared. The result of the application of the algorithm is known as the *ciphertext*.

An encryption scheme's security depends on the mathematical properties of the algorithm used, the soundness of the engineering and methodology by which it is implemented, and the length and handling of the key. Assuming that the math, the engineering, and the implementation are sound, in operational use security should require only that the key be kept secret. This is known as Kerckhoffs's Principle[8], or in a slightly different form, as Shannon's Maxim.[9]

# 3 Evolutionary and Revolutionary Threats to Encryption

Unfortunately, it's not easy to ensure that an encryption scheme is secure. A scheme can fail in a number of ways: The key length can be short enough so that it is (relatively) easily found; there can be vulnerabilities in the encryption algorithm, protocol, or implementation; or advances in technology can render a scheme insecure.

## 3.1 Key Length

If an algorithm has a key space that is too small, it's possible that the key can be found by an exhaustive key search, or "brute force attack." Such an attack tests every possible key until it finds the right one. For example, in 1998, Cryptography Research, Inc, Advanced Wireless Technologies and the Electronic Frontier Foundation (EFF) built a system called the EFF DES Cracker (nicknamed "Deep Crack") to demonstrate that the Data Encryption Standard (DES) algorithm's 56-bit key space was inadequate to provide useful security. Deep Crack used 1,856 custom application specific integrated circuit (ASIC) on 29 circuit boards of 64 chips each. The boards were mounted in six cabinets within a Sun-4/470 chassis. The machine was capable of testing over 90 billion keys per second. As a result, every possible key could be tested in about nine days. On average, the key would be found in about four and a half days.[10] Today, Crack.sh offers a system with 48 Xilinx Virtex-6 LX240T field programmable gate arrays (FPGA). Each FPGA contains a design with 40 fully pipelined DES cores running at 400MHz for a total of

---

[6] The Yahoo, Office of Personnel Management, Anthem, Equifax, Home Depot and Heartland Payment Systems breaches are examples of this.

[7] This is in contrast to *steganography*, which is focused on hiding a message's existence.

[8] http://www.crypto-it.net/eng/theory/kerckhoffs.html

[9] "The enemy knows the system."

[10] In contrast, if a supercomputer that performs at a peak speed of 10.51 Petaflops (Flop = Floating Point Operations Per Second), or 10.51 x $10^{15}$ Flops, and one possible key can be checked with 1000 flops (optimistic), a a 128-bit key (which has 3.4 x $10^{38}$ possibilities) will require one billion billion (1,000,000,000,000,000,000) years to be brute-forced. For comparison, the universe is only 13.75 billion years old.

16,000,000,000 keys/sec per FPGA, or 768,000,000,000 keys/sec for the whole system. This means that it can exhaustively search the entire 56-bit DES keyspace in about 26 hours.[11]

## 3.2 Advances in Cryptanalysis

Additionally, cryptanalysts (people who specialize in finding weaknesses in cryptosystems) discover cryptosystem vulnerabilities on a regular basis. For example, the RC4 stream cipher had been widely adopted, most notably in early WiFi protocols such as Wired Equivalent Privacy (WEP)[12] and the original (pre-WPA2) WiFi Protected Access (WPA).[13] It was also incorporated into the Web communications security protocols Secure Sockets Layer (SSL)[14] and its successor, Transport Layer Security (TLS)[15] [16]. By 2001, researchers discovered a number of vulnerabilities[17] with RC4 that ultimately led to the recommendation against its use.[18] Unfortunately, for a considerable period of time (years!) a very large percentage of the Internet continued to use insecure protocols that incorporated RC4.

## 3.3 Quantum Computing

Quantum computing takes advantage of the ability of subatomic particles to exist in more than one state at a time, which allows operations to be done much more quickly (and, theoretically, using less energy) than classic computers. Quantum computers manipulate what are called quantum bits or *qubits*. Unlike bits, which can exist in one of two states (i.e., and 1 or a 0), qubits can exist in a superposition of both states at the same time. For example, in classic computing, three bits can represent any one of eight values at a given point in time:

- 000
- 001
- 010
- 011
- 100
- 101
- 110
- 111

In quantum computing, three qubits can represent all eight values at the same time. A qubit can store much more information that a classical bit, and a quantum computer can perform algorithms that classical computers cannot. This has significant impacts on cryptography. For example, there are certain

---

[11] https://crack.sh/
[12] http://standards.ieee.org/getieee802/download/802.11i-2004.pdf
[13] http://standards.ieee.org/getieee802/download/802.11i-2004.pdf
[14] https://tools.ietf.org/html/rfc6101
[15] https://tools.ietf.org/html/rfc4346
[16] https://tools.ietf.org/html/rfc5246
[17] As a stream cipher, RC4 takes a short (e.g., 128-bit) key and stretches it into a long string of pseudo-random bytes. These bytes are XORed with the plaintext to create the ciphertext. Unfortunately, the bytes coming out of RC4 aren't really random. They have small but significant biases. As a result, successive encryptions of the same message with different RC4 aren't random. An analysis of different portions of the encrypted message will indicate that some values occur more often than others. An attacker that obtains a sufficient number of encryptions of the same message using different keys can, based on the biases (deviations from random), recover the plaintext.
[18] https://tools.ietf.org/html/rfc7465

classes of mathematical problems that quantum computing solves significantly faster than what is known for classical computers. These include computing discrete logarithms and factoring large integers. Unfortunately for modern cryptography, the assumed difficulty of these problems is the foundation on which asymmetric encryption is built. Solving those problems rapidly (e.g., using Shor's algorithm[19]) effectively compromises current abilities to conduct any online activity securely.[20] Quantum computing also impacts the symmetric algorithms commonly used to ensure data confidentiality; fortunately, that impact can be mitigated by increasing the key size for symmetric encryption algorithms.

# 4 Cryptographic Security Assurances

Given the many threats to data security, a brief discussion of the data security guarantees provided by symmetric ciphers is useful. Generally, a symmetric encryption scheme can provide two types of assurances, those relating to *confidentiality* and those relating to *authenticity* (also called integrity).

## 4.1 Confidentiality

The primary goal of an encryption scheme is to provide confidentiality for encrypted messages. Confidentiality can be rigorously defined in multiple ways; two of which are discussed here. In both cases, the goal is computational security rather than information-theoretic security. That is, although an adversary may theoretically break a scheme by enumerating all possible keys, it should be infeasible for an attacker to do so in any reasonable amount of time given available computing power. We are ultimately interested in quantifying precisely how difficult it will be for an attacker to break some scheme.

The strongest level of confidentiality an encryption scheme can provide is that it leaks *no information whatsoever* about the underlying message. This is typically formalized using the notion of *indistinguishability*. Roughly, an encryption method is *indistinguishable* if an adversary given the encryption of a message, chosen from one of two possibilities, will be unable to guess which message was encrypted with better than 50% probability. Moreover, this should hold even if the attacker has observed many previous encryptions (using the same key) of other messages.

A weaker, but still meaningful, notion of confidentiality for an encryption scheme is that it prevents *message-recovery attacks*. This, informally, means that an attacker given the encryption of a completely unknown message should be unable to recover the entire message.

## 4.2 Authenticity

When encryption is used to exchange information, such as in an online application, attackers are afforded the opportunity to intercept, tamper with, and then forward the altered encrypted messages to the intended recipient. Such attacks can cause many encryption schemes to fail, allowing attackers to completely recover the original messages. In some cases, attackers need only an error message from the receiver. In others, all that is needed is to measure the time it takes for the receiver to acknowledge

[19] https://en.wikipedia.org/wiki/Shor%27s_algorithm

[20] Not to worry, industry, academia and government organizations are working diligently to solve the problem and there is promising research being conducted in the fields of lattice-based cryptography, code-based cryptography, multivariate polynomial cryptography, hash-based signatures and isogenies on supersingular elliptic curves.

the submission. This is known as a *chosen-ciphertext attack*, and the most common version is the *padding oracle attack* discovered in 2002 by Serge Vaudenay.

To defend against these sorts of attacks, symmetric block cipher modes of operation characterized as *Authenticated Encryption* (AE) (sometimes *Authenticated Encryption with Associated Data* (AEAD)) are used. Such modes handle both encryption and authentication, thus assuring the recipient that the encrypted message received was the encrypted message sent and that there was no tampering. Some of the most common AE modes are Galois Counter Mode (GCM), Offset Codebook Mode (OCB), and Counter Mode with CBC MAC (CCM). In practice, these modes authenticate all of a message's components. GCM, for example, addresses authentication for the ciphertext contents and length, the initialization vector and the unencrypted header (the associated data) and its length.

# 5 PKMS2 Operational Overview

PKMS2 is a mode of operation that provides significant improvements in the degree of cryptographic assurance available from symmetric block ciphers such as the Advanced Encryption Standard (AES). Specifically, PKMS2:

- Relies on multiple ciphers, and provides "fallback security" in case one or more of those ciphers is broken by some future attack;
- Ensures indistinguishability with an effective key length larger than that of its constituent ciphers; and
- Uses segmentation to improve security against message-recovery attacks.

Importantly, PKMS2 achieves these improvements by incorporating trusted, well-researched symmetric block ciphers.[21] The current PKMS2 cipher suite includes the AES, Serpent, Speck, Simon, Aria, MARS, Camellia, and Twofish algorithms.

PKMS2 employs *layering* and *segmentation* to achieve its security improvements. A message being encrypted under the PKMS2 scheme is first segmented (or sharded) into 256 equally sized, contiguous segments. The actual size of each segment depends on the size of the original message or file. Padding is used to ensure that the last segment is of equivalent size.

[21] As per eminent cryptographer Bruce Schneier: "Anyone can design a cipher that he himself cannot break. This is why you should uniformly distrust amateur cryptography, and why you should only use published algorithms that have withstood broad cryptanalysis. All cryptographers know this, but non-cryptographers do not. And this is why we repeatedly see bad amateur cryptography in fielded systems." See: https://www.schneier.com/blog/archives/2015/05/amateurs_produc.html

A random 256-bit cryptographic key is used for each of the 256 segments, and one of the algorithms in the cipher suite is randomly allocated to each segment.[22] Each segment is then encrypted (using cipher block chaining (CBC) mode) using the specific key/algorithm combination. This creates the first encryption layer.

The resulting composite ciphertext is then subjected to a second round of segmentation and encryption using the same process, but with independent keys and ciphers chosen for each segment. (If fallback security is required, then two aligning segments are not allocated the same cipher.) At the conclusion of the second round, a table indicating the lengths of each segment and other metadata is concatenated with the resulting ciphertext. This table is referred to as the *pattern key*.

A final, encryption round is then conducted, encrypting the entire ciphertext previously created (along with the concatenated pattern key) using a single algorithm and key.

[22] In the Secure Channels ParaDoxBox implementation, the user is permitted to select which of the algorithms to use and may choose from one to all eight. There is a small increase in effective key size depending on how many ciphers are selected. It is assumed that the default will be to use all eight.

The specification for the PKMS2 mode of operation is deliberately silent on a number of issues, including:

- Key management; and
- Authenticity/authenticated encryption.

It is expected that implementations that integrate PKMS2 capabilities will provide for key management and authenticity guarantees. For example, the PKMS2 implementation in the Secure Channels ParaDoxBox product provides secure key management, and the final encryption round is conducted using AES operating in Galois Counter Mode (GCM) to provide the necessary authenticity guarantees.

# 6 Formal Analysis and Validation of PKMS2

A mathematical security analysis and cryptographic proof of PKMS2 was conducted by Dr. Jonathan Katz.[23] The analysis confirms the security claims about PKMS2 made earlier. Specifically:

- The analysis proves that "fallback security" holds. Specifically, even if a cipher used in the first layer is insecure (e.g., is eventually broken by some unforeseen attack), the encrypted message remains protected as long as the ciphers used in the second layer are secure.

- Assuming all the ciphers used in both layers are secure, the effective key length of the PKMS2 scheme is provably larger than the key length of any of the constituent ciphers. For example, assuming that the ciphers used have 256-bit keys and 128-bit block sizes (e.g., AES-256), this allows PKMS2 to obtain an effective key length of at least 387 bits.

- Security against message-recovery attacks (with respect to the best-known attacks) improves due to the use of segmentation.

Dr. Katz's proof is available upon request.

[23] Dr. Katz is the Director of the Maryland Cybersecurity Center and a Professor in the Department of Computer Science at the University of Maryland. He is a co-author of "Introduction to Modern Cryptography," a widely used textbook, and Vice President of Cryptographic Engineering at Secure Channels, Inc.

Dr. Katz's analysis and proof was reviewed, analyzed, and validated by Dr. Yevgeniy Dodis[24], Dr. Matthew Green[25] and Dr. Stefano Tessaro[26]. Their analyses are available upon request .

# 7 References


Dodis, Yevgeniy. *Proof Verification for PKMS2 Scheme*.

Green, Matthew. *Validation of a Security Proof of PKMS2*.

Katz, Jonathan. *Security Analysis of PKMS2*

Tessaro, Stefano. *Review of PKMS2's Security Analysis*.


[24] Dr. Dodis is a Professor of Computer Science at New York University's Department of Computer Science and the Courant Institute of Mathematical Sciences.

[25] Dr. Green is an Assistant Professor at the Johns Hopkins Information Security Institute and one of the creators of the Zerocash protocol, which is used by the ZCash cryptocurrency.

[26] Dr. Stefano Tessaro is an Assistant Professor in the Department of Computer Science at the University of California, Santa Barbara, where he currently holds the Glen and Susanne Culler chair.