AgilePQ DEFEND Cryptographic Tests

This document describes the cryptographic properties of AgilePQ DEFEND, and provides results to support the claim.

Agile and, Post-Quantum

Professors in the Mathematics and Computer Science Department at the University of California – San Diego (UCSD) reviewed the cryptographic strength of our algorithms, and after months of deliberation came to the conclusion that there are no known attacks against AgilePQ DEFEND.

This report will cover the mathematical strength of the algorithms at a high level, demonstrate the confusion and diffusion properties of AgilePQ DEFEND, compare these properties against AES256-CBC to show they are equal or better, and discuss the results of the DieHarder Randomness Testing Suite, a standard statistical testing suite from NIST.

The results show that the output of AgilePQ DEFEND resembles that of a cryptographically secure random number generator, and no discernible patterns or vulnerabilities exist in AgilePQ DEFEND. Additionally, we find that the cryptographic properties of AgilePQ DEFEND are equivalent to or better than AES256 and AES128, and we accomplish this significantly faster and with less overhead than AES. The power consumption testing (Power Comparison White Paper) backs up our results, and we find that AgilePQ DEFEND is an efficient, fast, and strong encryption technology with an enormous range of applicability.

Overview

We compare AgilePQ DEFEND to AES256 and traditional cryptography. This report details the testing and measurement surrounding the cryptographic properties of AgilePQ DEFEND.

For any data protection solution – there are four key elements to a secure encryption technology:

- Key Search Space
- Ensuring no Repetition
- Confusion (Randomness)
- Diffusion (Change)

On these elements – AgilePQ DEFEND compares as follows:

- AgilePQ DEFEND Key Search Space is 429 orders of magnitude larger than AES
- AgilePQ DEFEND is much better for ensuring no repetition due to continuously changing key tables
  - AES will produce the same encoded output for the same input
- Both AES & AgilePQ DEFEND are highly random (follow a Gaussian distribution)
- Both AES & AgilePQ DEFEND produce good Diffusion

Difficulty And Key Space

AgilePQ uses a novel key technique implemented by tables. These tables may vary in size between applications, e.g., memory bound Internet of Things devices can use smaller tables and packet sizes than Ethernet. On average, the table has 256 elements, but the table size can be adapted to any size. This table is a secret, symmetric key between two communicating parties. AgilePQ DEFEND also employs an additional secret of 28 bytes as an Initialization Vector (IV) alongside the tables. This secret is established in a secure manner using well-known cryptographic key exchange methods.

AES256-CBC uses a single key of 256 bits, and an initialization vector of 16 bytes (128 bits). The AES IV is public, and only the 256-bit key is kept secret. Because AES256 uses the public initialization vector, the secret key space of AES is 2^256, calculated as the number of possible 256-bit binary values, which is approximately equal to 1.157×10^{77}.

AgilePQ DEFEND has a secret key space of 256 factorial, or 256!, which is approximately equal to 8.578×10^{506}. This is a 429 order of magnitude difference between AgilePQ DEFEND and AES256. This shows that AgilePQ DEFEND far exceeds the brute-force difficulty of AES256.

Furthermore, the Ponemon Institute, The University of New South Wales, The University of California – San Diego, and others have validated AgilePQ DEFEND's cryptographic strength. The validations include red teaming by skilled and determined teams in an attempt to break the code and in depth mathematical analysis by recognized cryptographic experts.

Repeated Message Encryption

This section illustrates one key difference between AES256 and AgilePQ DEFEND – that of ensuring no repetition between encrypted messages. Two plots are shown in the plots below representing the encrypted message as 16 bit numbers. The blue line represents one encryption of the message. The orange line is that same message encrypted a second time. Both times the algorithms use the same initialization. The x-axis is the time the message is received, and the y-axis is the 16-bit value of the message. The messages are 256-bits in this test, and are converted to a series of 16-bit numbers for plotting purposes. The first plot shows the output from AgilePQ DEFEND.

AgilePQ DEFEND

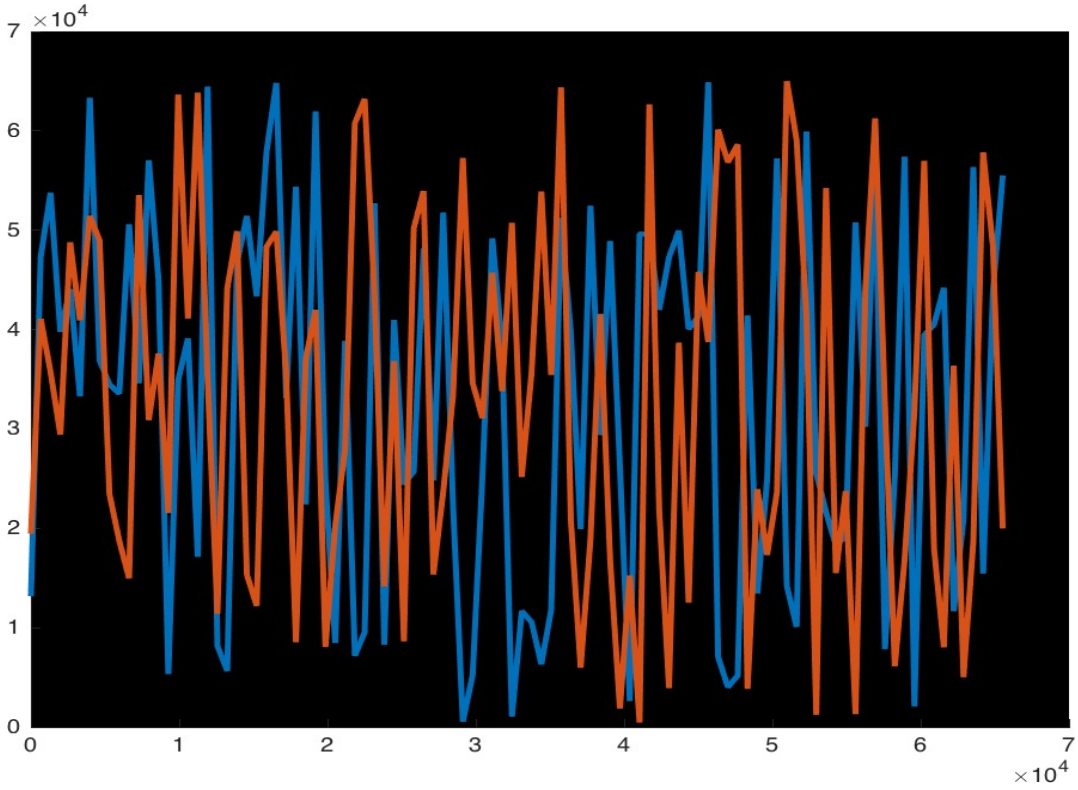

FIGURE 1: Same Message Encrypted Twice With AgilePQ DEFEND

Figure 1 on page 3 (previous page) shows that AgilePQ DEFEND avoids repeating cipher-texts for the same message, with the initialization remaining the same between encryptions. The blue line and orange line show that no correlation exists between these two encrypted messages, thus semantic security is preserved.

Figure 2 below shows the output from AES256. In this case the orange line and blue line are exactly superimposed.

AES256

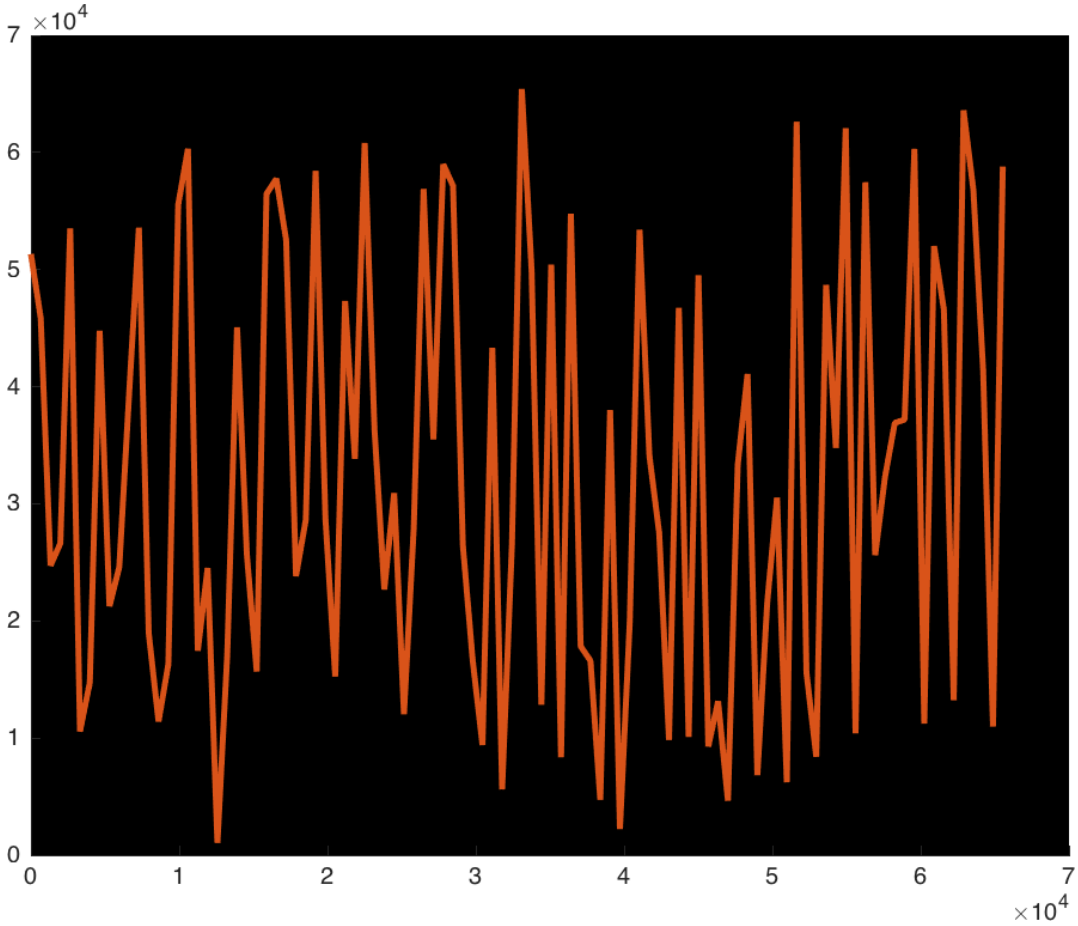

FIGURE 2: Same Message Encrypted Twice With AES

In AES, the same initialization is used for both tests as it was for the AgilePQ DEFEND test. It is noted that with AES the same message encrypted twice results in the same cipher-text output. Only in certain implementations of AES will the same message not encrypt to the same cipher-text.

This is apparent in the figure above, where the blue and orange lines overlap exactly. AgilePQ DEFEND does not suffer from this weakness because of how it is designed and is one of the differentiating factors between AgilePQ DEFEND and current encryption technology such as AES256.

Confusion

This section analyzes the randomness of AgilePQ DEFEND and AES256 against a random (Gaussian) distribution.

Comparing AgilePQ DEFEND and AES256 in this manner shows that both AES and AgilePQ DEFEND offer great confusion of the input, producing an output that is indiscernible from true randomness.

For each of these tests, the parameters are the same:

INPUT	The quick brown fox jumps over the lazy dog.
ITERATIONS	128,000
RANDOM NUMBER GENERATOR(blue)	/dev/urandom
PROGRAM	MATLAB R2016a

AgilePQ DEFEND

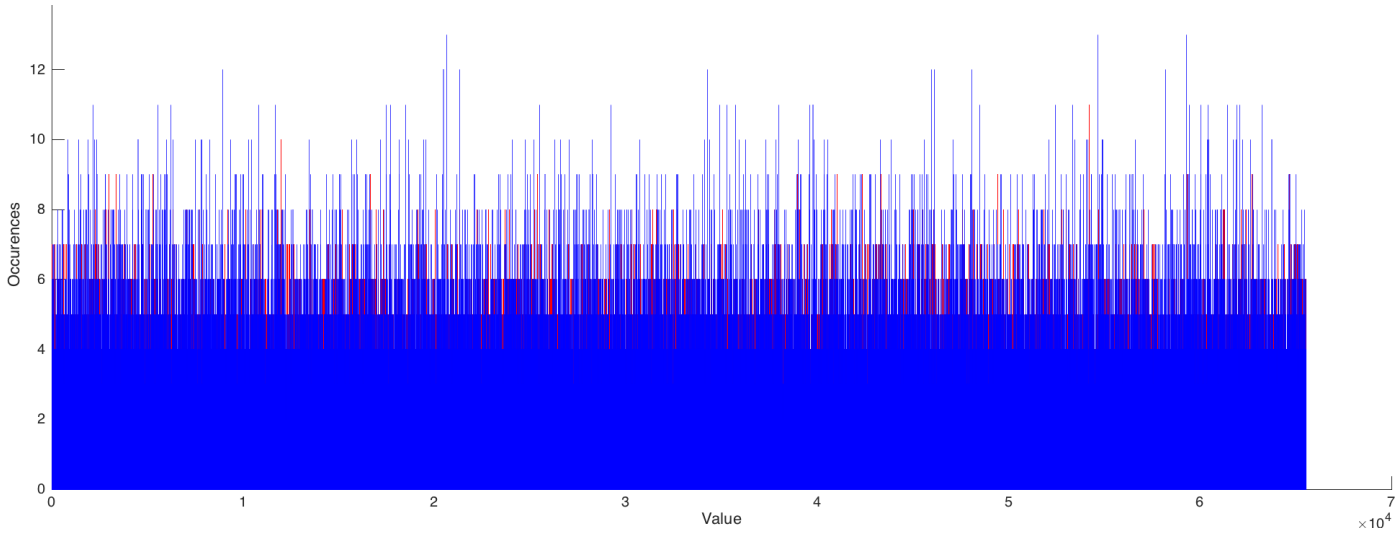

*FIGURE 3: **Histogram** AgilePQ DEFEND (red) and Random Number Generator (blue)*

Figure 3 (see above) is a histogram of a sampling of AgilePQ DEFEND outputs, plotting the value that occurs on the X axis, and the number of times it occurs on the Y axis. We notice that the data is equally and randomly distributed across all bins, with no bias towards a collection of values, and few high count occurrences. AgilePQ DEFEND occurs in red, and a random Gaussian Distribution is in blue. The Gaussian Distribution is generated using MATLAB's randomness feature, and is a good standard against which to compare. We see that AgilePQ DEFEND would be indistinguishable from random noise if the colors were the same.

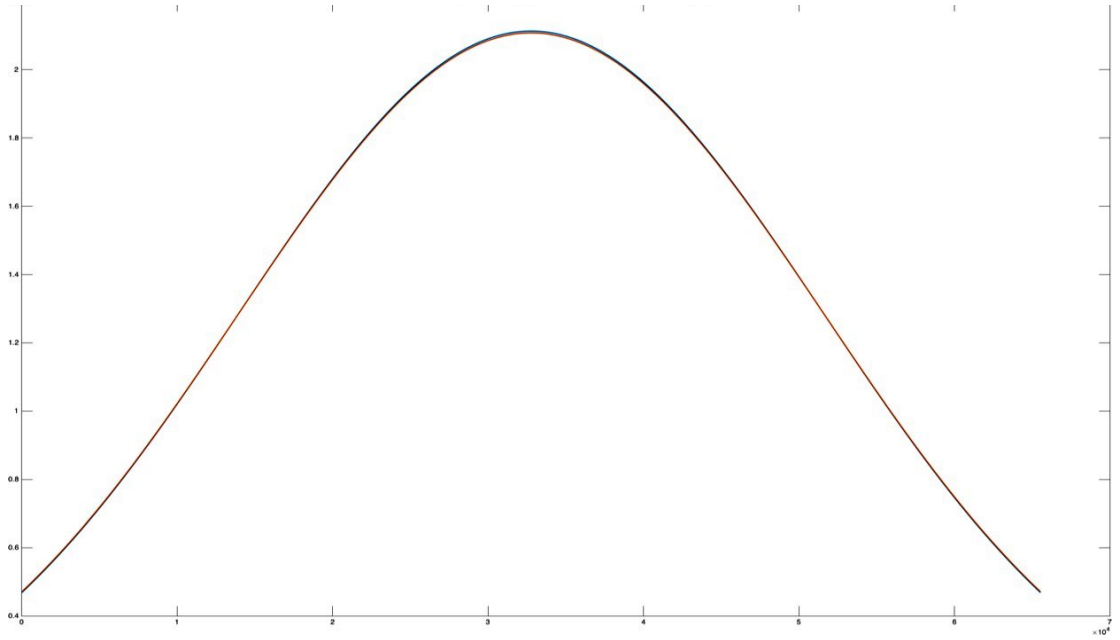

*FIGURE 4: **Gaussian** AgilePQ DEFEND (red) and Random Number Generator (blue)*

Figure 4 shows the probabilistic distribution of AgilePQ DEFEND output, and that of a random number distribution. Because AgilePQ DEFEND output is “random”, or normal, its probability distribution function is a Gaussian (normal) distribution.

AES256

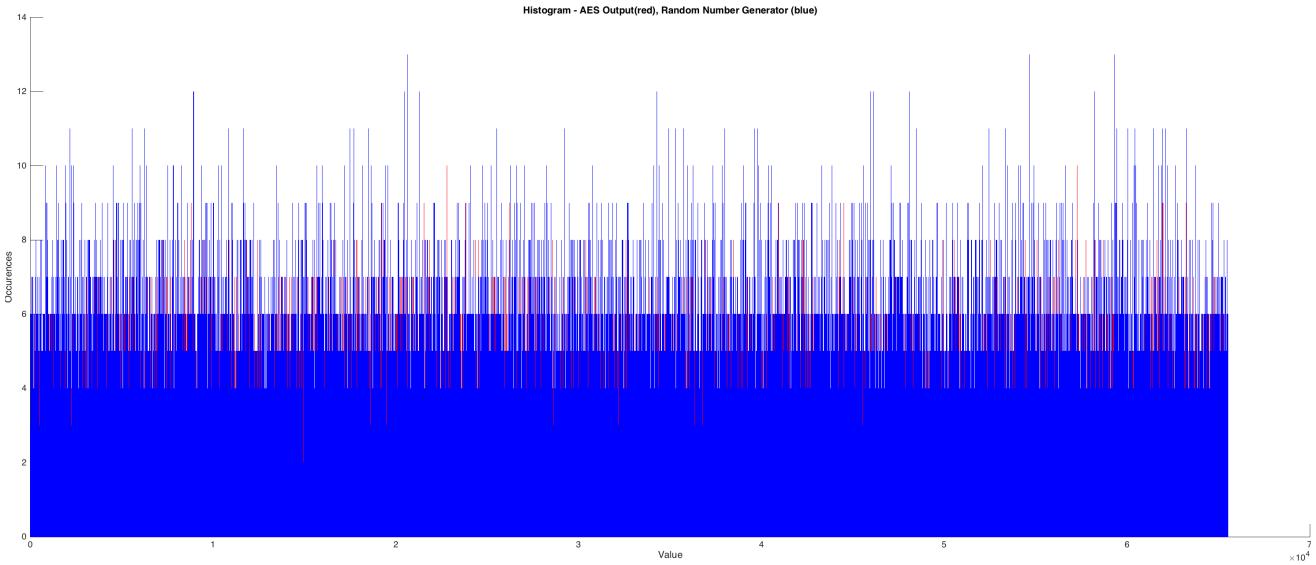

*Figure 5: **Histogram** AES256-CBC (red) and Random Number Generator (blue)*

Similarly, in figure 5 we can see that AES256 shows good confusion in its output as evidenced in this Histogram.

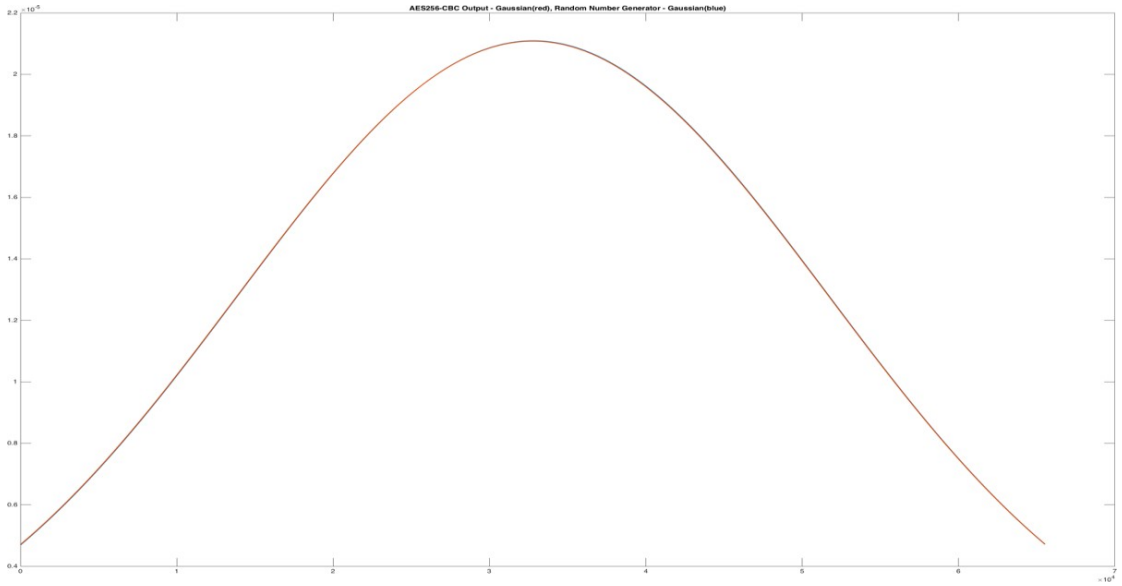

FIGURE 6: ***Gaussian*** *AES256-CBC (red) Random Number Generator (blue)*

In figure 6 (see above) we can see that AES256 also follows a Gaussian Distribution, showing that AgilePQ DEFEND and AES are both sufficiently strong cryptographic algorithms.

Diffusion

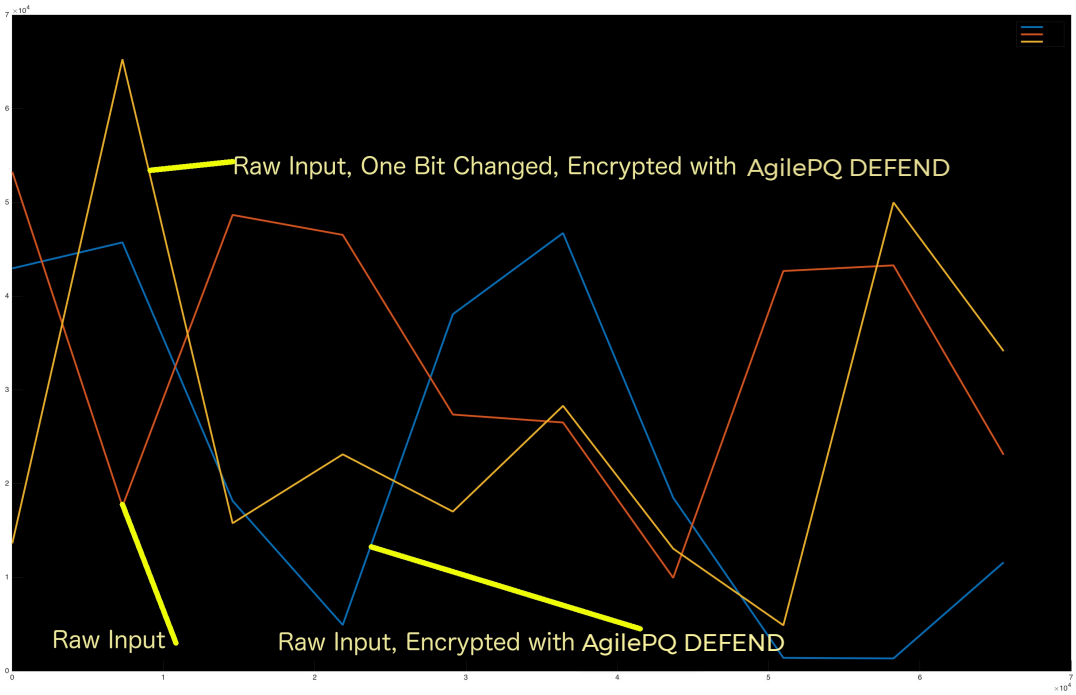

Figure 7: AgilePQ DEFEND(blue) applied to raw input (red), and AgilePQ DEFEND applied to the same input with 1 bit changed (yellow)

Figure 7 (see above) shows the cryptographic Diffusion property of AgilePQ DEFEND. A random input is given to AgilePQ DEFEND; this input is plotted in red. AgilePQ DEFEND encrypts this input, and the values of the encrypted message are shown in blue. Next, a single bit is changed in the input (not pictured), and AgilePQ DEFEND encrypts this next message with the same key. The resulting values are seen in the yellow line.

This shows that AgilePQ DEFEND properly diffuses a single bit change into the resulting cipher-text, preventing attackers from obtaining information about the key by modifying input messages by a single bit.

A good encryption algorithm should have approximately 50% diffusion after a single iteration, and both AgilePQ DEFEND and AES accomplish this. We calculate the diffusion percentage in the following manner, on a single iteration of both AgilePQ DEFEND and AES.

Compute difference % in output with AES and AgilePQ DEFEND:

- Generate 16 bit input **x**
- Apply algorithm to obtain output y_1
- Change single bit in **x**
- Apply algorithm to obtain output y_2
- Bitwise XOR $y_1 \oplus y_2 = d$
- There should be an approximate 50% distribution of 1's and 0's in **d**

For AgilePQ DEFEND:

We calculate the diffusion of the AgilePQ DEFEND according to the algorithm above. The values we obtain are y_1 = 53252, y_2 = 13174.

53252 ⊕ 13174 = 58226 = 1110001101110000

Of the 16 bits, 8 are 1's, and 8 are 0's, showing an exactly 50% diffusion.

For AES256:

For AES, we calculate the diffusion in the same manner. The values we obtain are y_1 = 39329, y_2 = 53575.

39329 ⊕ 53575 = 18662 = 100100011100110

Of the 16 bits, 7 are 1's, and 9 are 0's, showing 43.75% diffusion.

This is only one sampling, but the diffusion should be at or around 50% between any two chosen values in the cipher-text.

DieHarder Randomness Test

These tests were run using the DieHarder Testing Suite from Duke University, found here.

The DieHarder Testing Suite is for testing statistical properties of random number generators. While the AgilePQ DEFEND and AES256 are not necessarily random number generators, an encryption algorithm should have strong statistical randomness properties to maintain semantic security, also referred to as "confusion of the data". A weak or failed result does not imply that the corresponding algorithm is weak or broken. These results reinforce the confusion results above showing that AgilePQ DEFEND and AES are equivalently strong in their randomness properties.

Baseline: /DEV/URANDOM

- Passed **65**
- Failed **2**
- Weak **3**

70 Tests Completed

We use /dev/urandom as a baseline randomness test, as it uses non-deterministic "true" randomness from the computer. /dev/urandom is a preferred source for cryptographic random number generation on Unix systems. We see that it fails two tests, and is "statistically weak" in 3. This does not mean that it is a weak or vulnerable random number generator, it simply means that it did not pass those tests with statistical significance.

The same tests were run against AgilePQ DEFEND and AES with these results:

AgilePQ DEFEND

- PASSED **66**
- FAILED **0**
- WEAK **4**

70 tests completed

AES256

- PASSED **68**
- FAILED **0**
- WEAK **2**

70 tests completed

We see that AgilePQ DEFEND and AES both perform as well or better than a cryptographically secure random number generator. AES is widely known to be a good random number generator, but it does this at a significant overhead of speed, processing requirements, and power consumption. AgilePQ DEFEND, on the other hand, performs as well as AES with a fraction of the overhead.

The "weak" tests in AES and AgilePQ DEFEND are those that /dev/urandom performed weakly or failed outright. These "weak" tests do not open up attack surfaces or opportunities to reverse the encryption. We can conclude that both AES and AgilePQ DEFEND are strong cryptographic random number generators and encryption algorithms, but AgilePQ DEFEND accomplishes this faster and with significantly less overhead than AES.

Conclusion

AgilePQ DEFEND has been validated both internally by running tests and measurements across a wide range of real-world devices, and externally through mathematical analysis, real-world red team attacks, and performance measurements by third parties. These results have consistently shown that the AgilePQ DEFEND is as strong as AES, and accomplishes this in a fraction of the time and power consumption.

We see that AgilePQ DEFEND and AES256 both have strong cryptographic properties, AgilePQ DEFEND outperforms AES256 in repeated encryption, and that AgilePQ DEFEND has a key space 429 orders of magnitude greater than AES256. This, in addition to the power and load testing results, show that AgilePQ DEFEND is an outstanding cryptographic alternative for the 21st century, especially in the growing number of resource constrained devices.

If you would like to test AgilePQ DEFEND, or for any further information, please contact:

Greg Ward (VP Product Strategy)
Email: gward@agilepq.com