

V O R T E X

*A New Family of one Way Hash Functions Based on Rijndael
Rounds and Carry-less Multiplication*

Shay Gueron^{1,2} and Michael Kounavis¹

¹Intel Corporation

²University of Haifa, Israel

Performance

Implementation	Vortex 224 (cycles/byte)	Vortex 256 (cycles/byte)	Vortex 384 (cycles/byte)	Vortex 512 (cycles/byte)
Reference (64bit)	46.46	46.46	61.67	61.67
Optimized 64-bit	46.26	46.26	56.05	56.05
Optimized 32-bit	69.44	69.44	90.07	90.07
Assembly (stand-ins)	2.47	2.47	2.22	2.22

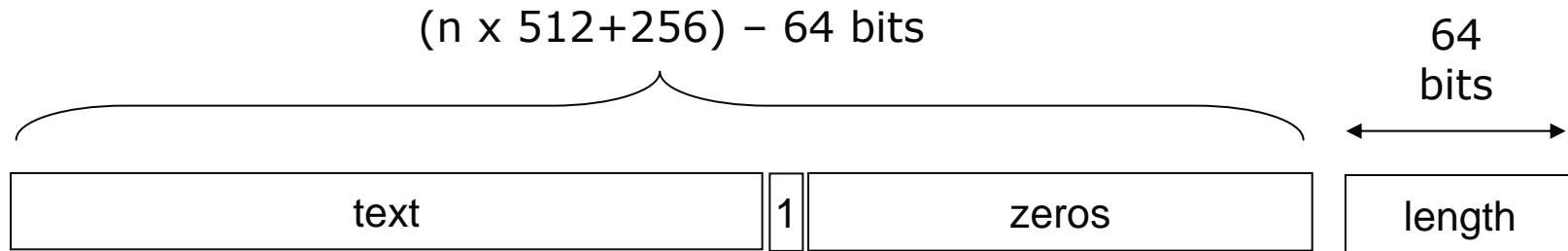
- Using Intel's new AES-NI, PCLMULQDQ instructions

The Design Philosophy

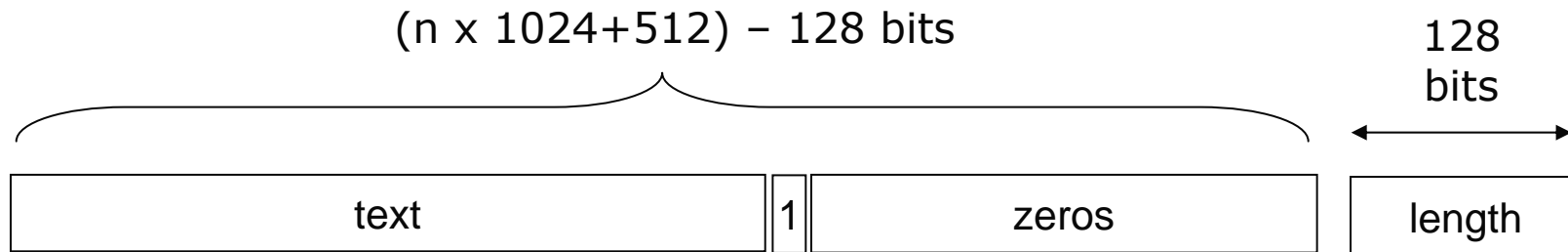
- **Domain Extension Transform:**
 - enveloped Merkle-Damgård
- **Uses the Rijndael round as a building block**
 - Rijndael round does good mixing
 - high performance due to new AES instructions
 - **Trend** in the industry: such functions supported by **dedicated hardware**
- **New method for merging 128/256-bit blocks into a longer digest**
 - new mode of operation
 - merging is non-commutative
 - take advantage of carry-less multiplication (PCLMULQDQ) instruction
- **Relation to AES-based hashes**
 - stronger key schedule algorithm
 - variant number of rounds
 - balances cryptographic strength

Block Length and Padding

Vortex 224, Vortex 256



Vortex 384, Vortex 512

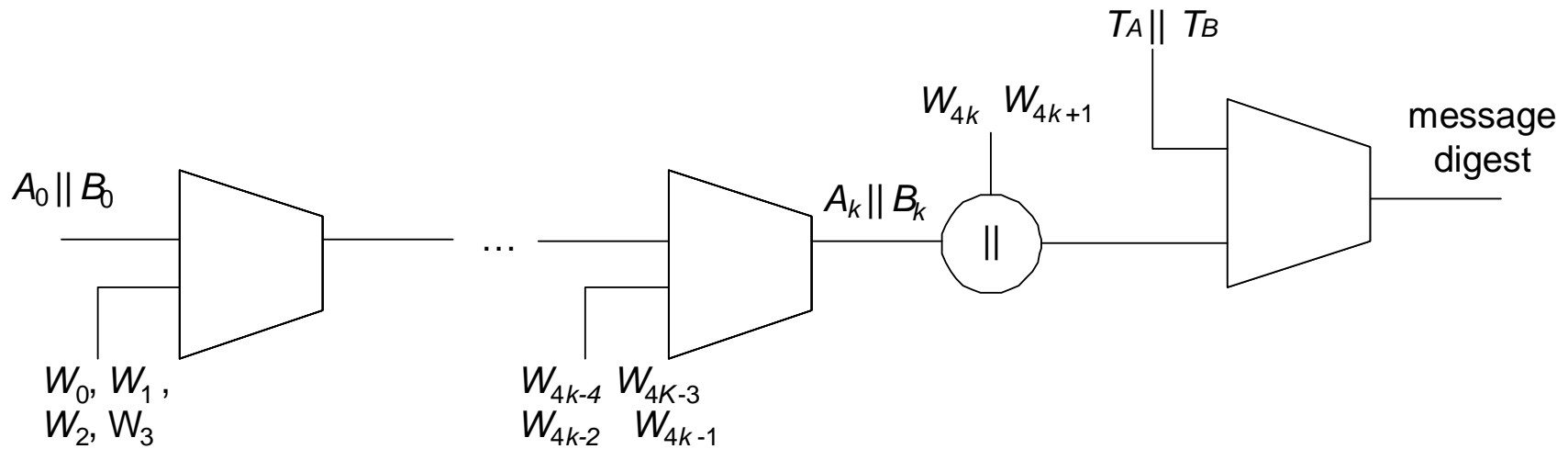


Tunable Parameters

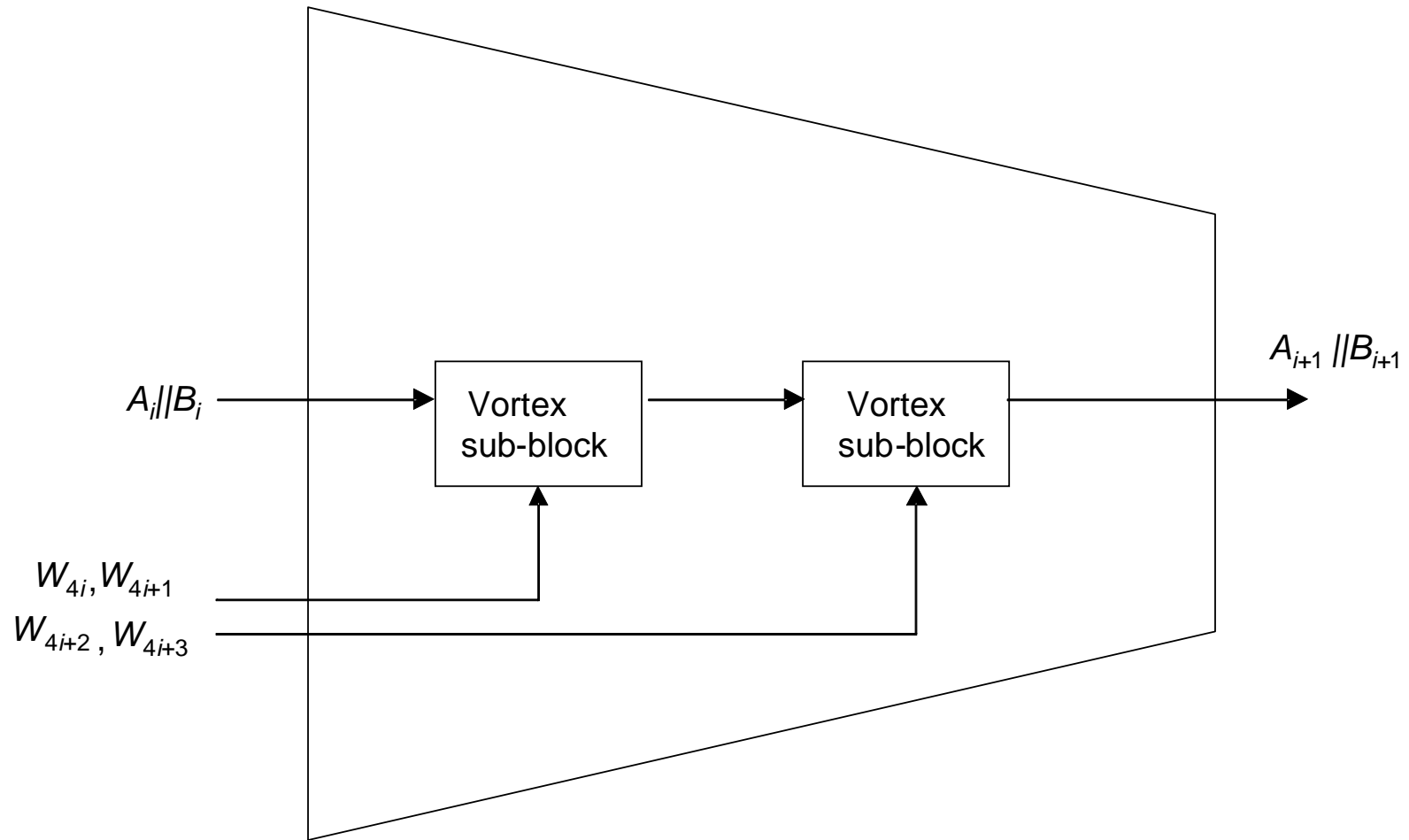
- Number of rounds N_R
 - specifies how many rounds are executed in the internal block cipher component of Vortex
- Degree of Diffusion D_F
 - determines how many times a bit is diffused over all bits of the output digests
- Multiplication type M_T
 - integer or Carry-less
 - determines the ease of quantitative analysis vs. non-linearity of the hash
- Initial value of the chaining variable $A_0//B_0$ and final tweak $T_A//T_B$
 - if parameterized, define a pseudorandom function family

Domain Extension Transform

- Enveloped Merkle-Damgård to support pseudo-random function and pseudorandom oracle preservation

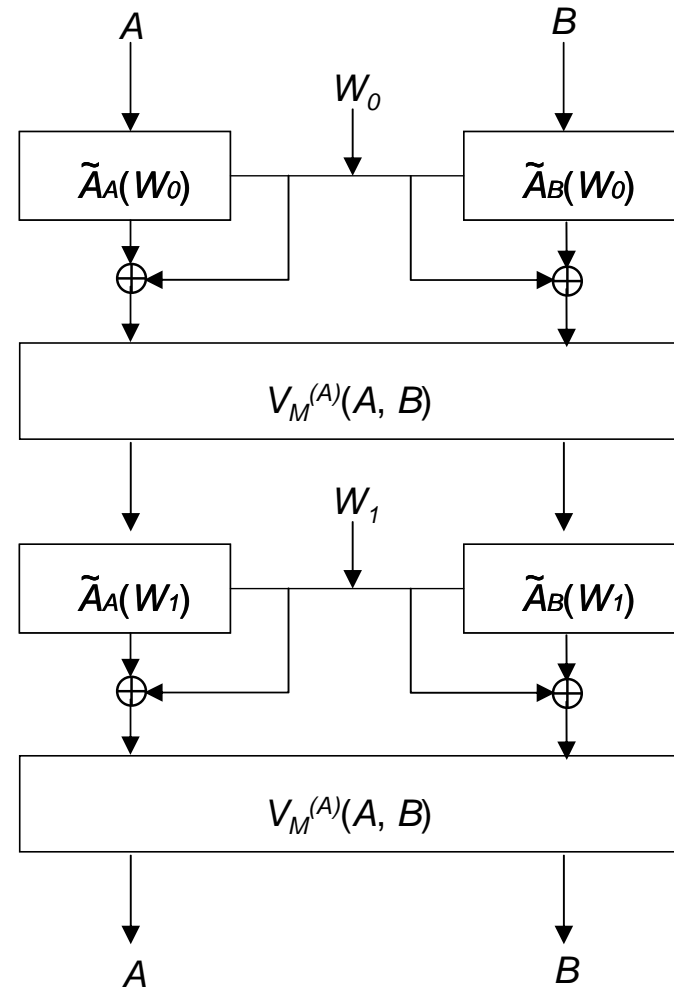


Vortex Block



Vortex Sub-block

- Variant of MDC-2
- Uses Matyas-Meyer-Oseas to avoid related key attacks
 - concept also used in other designs



A-Rijndael Transformation

Input: 128 (256)-bit block (B) and 128 (256)-bit key (K)

Expand K (128 (256)-bit) to m round keys RK[1], RK[2], ... RK[m] (using modified key generation)

For j from 1 to m do

B = A_RIJNDAEL_Encrypt_Round (B, RK [j])

End

The modified key generation scheme

RK[1] = K

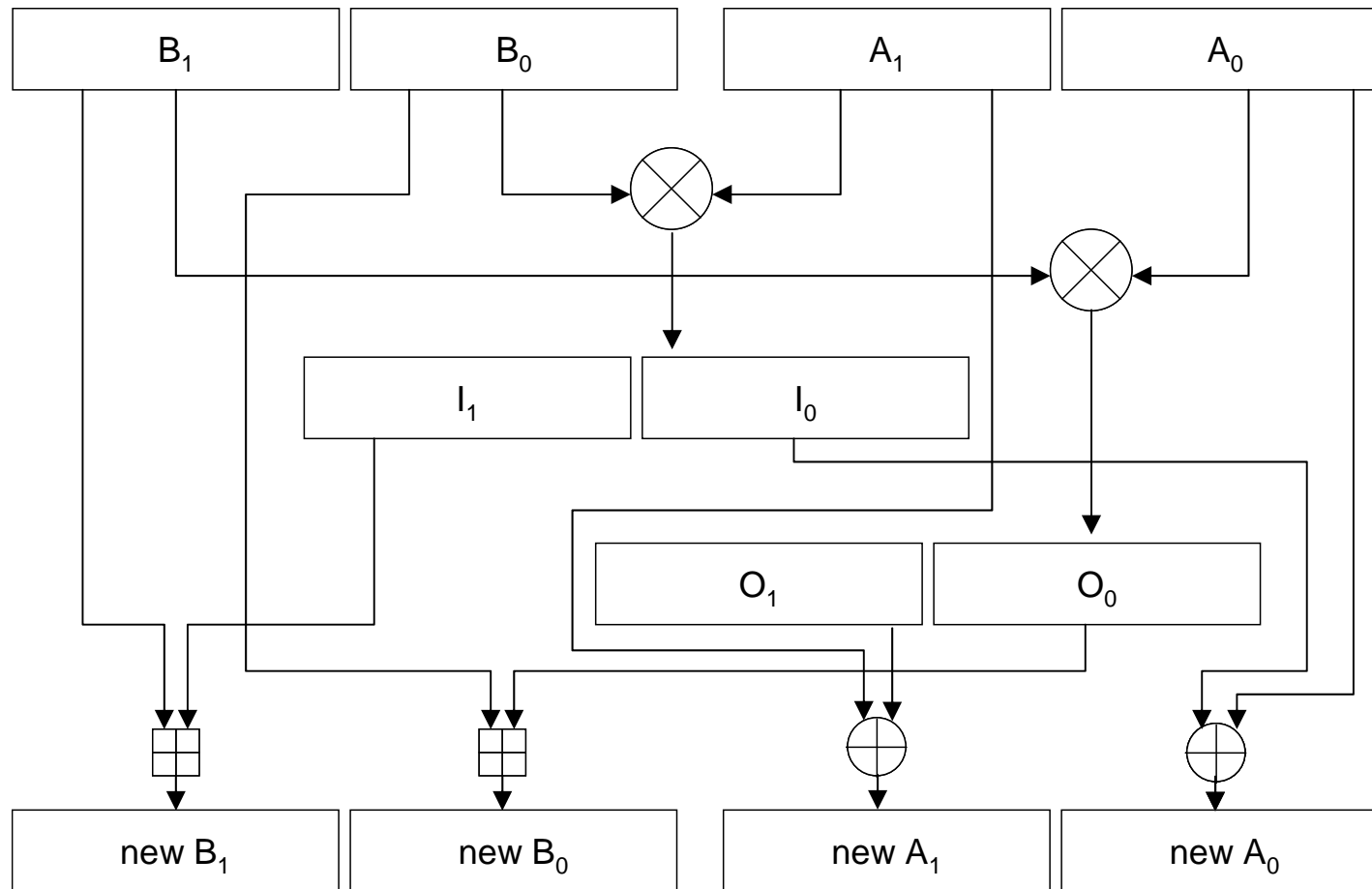
For j from 2 to m do

RK[j] = Perm(S-Box16 (32)(RK[j-1] + RCON [j]))

End

+ = integer addition mod 2^{64} (knock off final carry in every 64 bits)

Merging Function



Last Vortex Sub-block

Last Vortex sub-block(A, B, W_0, W_1, D_F)

begin

; W_0 is the first word of the last sub-block to be processed

$$A \leftarrow \tilde{A}_A(W_0) \oplus W_0$$

$$B \leftarrow \tilde{A}_B(W_0) \oplus W_0$$

$$A \parallel B \leftarrow V_M^{(A)}(A, B)$$

for $i \leftarrow 1$ to $D_F - 1$ **do**

D_F is the degree of diffusion

W_1 is the second word of the current sub-block to be processed

$$A \leftarrow \tilde{A}_A(W_1) \oplus W_1$$

$$B \leftarrow \tilde{A}_B(W_1) \oplus W_1$$

$$A \parallel B \leftarrow V_M^{(A)}(A, B)$$

;
;
;

$$A \leftarrow \tilde{A}_A(W_1) \oplus W_1$$

$$B \leftarrow \tilde{A}_B(W_1) \oplus W_1$$

return $A \parallel B$

}

Performance Analysis

Implementation	Vortex 224 (cycles/byte)	Vortex 256 (cycles/byte)	Vortex 384 (cycles/byte)	Vortex 512 (cycles/byte)
Reference (64bit)	46.46	46.46	61.67	61.67
Optimized 64-bit	46.26	46.26	56.05	56.05
Optimized 32-bit	69.44	69.44	90.07	90.07
Assembly (stand-ins)	2.47	2.47	2.22	2.22

- AES-NI, PCLMULQDQ trend in the industry, also int mul variant
 - Sun, IBM are researching the technology (Moriokah Satoh et. al., Eberle et. al.)
 - good instruction sets are widely adopted (SSE, SSE2, EM64T)
 - e.g., for video streaming, multimedia, graphics
 - embedded systems processors are likely to also support AES

Qualitative Analysis

Properties

- Rijndael round: Good mixing function
- block cipher keys come from the chaining variable
- two independent sources of non-linearities in the key schedule
- SBox(), adds with carries
- Matyas-Meyer-Oseas to avoid related key attacks
- non-commutative merging function

Issues

- some impossible images, output bit correlation
- Zero accumulation point in the multiplier
- result of carry-less multiplication is 127 (255) bits long (top bit=0)

So, we will omit the final call to the merging function

Theoretical Analysis

Theorem 1:

The number of queries required for finding a collision with probability greater or equal to 0.5 in an ideal block cipher approximation of A-Rijndael is at least $1.18 \cdot 2^{122.5}$ for Vortex 224 and Vortex 256, if the attacker uses randomly chosen message words for the queries.

collision resistance of at least
122.5 bits

Generic Attacks

- Algebraic
 - Can be mitigated by tuning:
 - Number of rounds, degree of diffusion, Perm()
- Related Key
 - Mitigated by using Matyas-Meyer-Oseas
- Multi-collision
 - Single collision complexity is high already (2^{122})
- Side Channel
 - AES round, GFMUL can be implemented using combinatorial logic
 - No lookup tables
- Birthday
 - Same message feeds two parallel A-Rijndael blocks

Others' Analyses

- *Lars R. Knudsen, Florian Mendel, Christian Rechberger, Søren S. Thomsen*
 - Collision and Preimage Attacks on Vortex as submitted to the SHA-3 competition
 - Time: 2^{192} , Space: 2^{70}
- *Jean-Philippe Aumasson, Orr Dunkelman*
 - A note on Vortex' security: identified 2^{120} impossible images
 - At complexity 2^{96} (currently at success prob. 2^{-135})
 - *We will skip last merging step*
- *Niels Ferguson*
 - Simple correlation on some of the output bits of Vortex
 - *We will skip last merging step*

Skipping Last Merging Step

$$A \leftarrow \tilde{A}_A(W_1) \oplus W_1$$

$$B \leftarrow \tilde{A}_B(W_1) \oplus W_1$$

$$A \parallel B \leftarrow V_M^{(A)}(A, B)$$

$$A \leftarrow \tilde{A}_A(W_1) \oplus W_1$$

$$B \leftarrow \tilde{A}_B(W_1) \oplus W_1$$

$$A \parallel B \leftarrow V_M^{(A)}(A, B)$$

...

$$A \leftarrow \tilde{A}_A(W_1) \oplus W_1$$

$$B \leftarrow \tilde{A}_B(W_1) \oplus W_1$$

~~$$A \parallel B \leftarrow V_M^{(A)}(A, B)$$~~

Summary

- **Main Result:**

- A collision resistant hash function (at least 122 bits of collision resistance)
- Vortex 256: 2.2 cycles per byte on next generation Intel CPU's
- Vortex 512: 2.5 cycles per byte on next generation Intel CPU's

- **Features**

- Supports 224, 256, 384, 512 message digests
- Based on Rijndael rounds, carry-less multiplication
- Uses EMD structure to preserve Pseudorandom Function/Oracle
- Taking advantage of new architectures

- **Attacks/Amendments**

- impossible images/output bit correlation/pre-image attacks can be addressed by skipping the last step

Backup

Useful Lemma

Lemma 1: Let $[W : Z] = X \text{ \AA } Y$ be the result of the carry-less multiplication of quantities X and Y defined as bit sequences and distributed uniformly. Then the probability that $[W : Z]$ takes a specific value $[\tilde{W} : \tilde{Z}]$ in the set $[0, 2^{128} - 1]$ is bounded by:

$$\Pr([W : Z] = [\tilde{W} : \tilde{Z}]) \leq 2^{-122.5}, \quad [\tilde{W} : \tilde{Z}] \in [0, 2^{128} - 1]$$

Moreover, the probability that Z takes a specific value \tilde{Z} in the set $[0, 2^{64} - 1]$ is bounded by:

$$\Pr(Z = \tilde{Z}) \leq 2^{-61.66}, \quad \tilde{Z} \in [0, 2^{64} - 1]$$

and the probability that W takes a specific value \tilde{W} in the set $[0, 2^{64} - 1]$ is bounded by:

$$\Pr(W = \tilde{W}) \leq 2^{-60.83}, \quad \tilde{W} \in [0, 2^{64} - 1]$$

Another Useful Lemma

Lemma 2: Let $[C : D] = \text{Query}(A, B, W_0)$ be the output of a query operation on $N/4$ bit quantities A, B, W_0 . Let $N=512$. Let also W_0 be uniformly distributed and the A-Rijndael transformation used by $\text{Query}()$ replaced by an ideal block cipher. Then the probability that $[C : D]$ takes a specific value $[\tilde{C} : \tilde{D}]$ in the set $[0, 2^{256} - 1]$ is bounded by:

$$\Pr([C : D] = [\tilde{C} : \tilde{D}]) \leq 2^{-245}, \quad [\tilde{C} : \tilde{D}] \in [0, 2^{256} - 1]$$

where by $\text{Query}()$ we mean
A-Rijndael+ Matyas-Meyer-Oseas + merging

Experimental Analysis: Hamming Weight Analysis (I)

Experiment	$N_R=3, D_F=5$	$N_R=5, D_F=5$	$N_R=7, D_F=5$	$N_R=10, D_F=5$
Short Messages, Vortex 224	112.1±7.4	111.8±7.5	112.3±7.6	112.0±7.4
Long Messages, Vortex 224	112.3±7.4	111.3±7.4	112.2±8.0	111.9±7.5
Short Messages, Vortex 256	128.1±8.1	127.8±8.0	128.3±8.1	128.0±7.8
Long Messages, Vortex 256	128.6±7.9	127.2±7.8	128.2±8.4	128.0±7.9
Short Messages, Vortex 384	191.8±9.8	192.2±9.9	192.3±9.8	192.1±9.9
Long Messages, Vortex 384	191.7±9.7	191.6±10.1	192.1±10.1	192.5±10.4
Short Messages, Vortex 512	255.7±11.3	256.2±11.5	256.1±11.4	256.0±11.4
Long Messages, Vortex 512	255.5±11.4	255.6±11.7	255.8±11.5	256.0±11.8

Experimental Analysis

Hamming Weight Analysis (II)

Experiment	$N_R=3, D_F=5$	$N_R=5, D_F=5$	$N_R=7, D_F=5$	$N_R=10, D_F=5$
Short Messages, Vortex 224	112.1±7.3	111.9±7.5	112.1±7.6	111.8±7.4
Long Messages, Vortex 224	112.5±7.4	111.9±7.7	112.6±7.3	111.9±7.9
Short Messages, Vortex 256	128.0±7.8	127.9±7.9	128.2±8.1	127.9±8.0
Long Messages, Vortex 256	128.3±7.9	128.0±8.4	128.5±7.7	127.9±8.3
Short Messages, Vortex 384	192.3±10.2	192.2±10.0	192.1±9.9	192.1±10.0
Long Messages, Vortex 384	191.4±10.2	191.8±9.4	192.7±10.1	191.1±9.9
Short Messages, Vortex 512	256.4±11.8	256.0±11.4	256.1±11.3	256.2±11.4
Long Messages, Vortex 512	255.1±12.4	255.7±11.3	256.4±11.6	255.0±11.0

Single-bit Differential Analysis

Scaled Mean Values ($M_T=0$)

Experiment	$N_R=3, D_F=5$	$N_R=5, D_F=5$
Vortex 256 (1 bit)	8192.37 ± 64.03	8191.93 ± 63.99
SHA 256 (1 bit)	8191.99 ± 63.96	8192.6 ± 63.98
Vortex 256 (16 bits)	53689.75 ± 238.16	53687.54 ± 242.44
SHA 256 (16 bits)	53689.86 ± 238.91	53693.23 ± 241.05
Experiment	$N_R=7, D_F=5$	$N_R=10, D_F=5$
Vortex 256 (1 bit)	8191.69 ± 64.08	8192.03 ± 63.98
SHA 256 (1 bit)	8191.75 ± 63.93	8191.99 ± 63.85
Vortex 256 (16 bits)	53688.50 ± 242.03	53685.16 ± 240.04
SHA 256 (16 bits)	53682.48 ± 246.06	53681.98 ± 242.25

Single-bit Differential Analysis

Scaled Mean Values ($M_T=1$)

Experiment	$N_R=3, D_F=5$	$N_R=5, D_F=5$
Vortex 256 (1 bit)	8192.46 ± 63.87	8191.81 ± 63.96
SHA 256 (1 bit)	8192.32 ± 63.92	8192.16 ± 64.22
Vortex 256 (16 bits)	53690.16 ± 245.51	53680.92 ± 239.86
SHA 256 (16 bits)	53690.73 ± 243.23	53689.34 ± 241.73
Experiment	$N_R=7, D_F=5$	$N_R=10, D_F=5$
Vortex 256 (1 bit)	8191.67 ± 63.91	8192.33 ± 64.24
SHA 256 (1 bit)	8191.96 ± 63.94	8191.76 ± 63.86
Vortex 256 (16 bits)	53678.77 ± 244.83	53687.57 ± 242.94
SHA 256 (16 bits)	53681.87 ± 242.01	53688.66 ± 243.16

Single-bit Differential Analysis Autocorrelation Matrix (Element Sum)

Experiment	$N_R=3, D_F=5, M_T=0$	$N_R=5, D_F=5, M_T=0$
Vortex 256 (16 bits)	147.99 ± 55.27	148.71 ± 55.90
SHA 256 (16 bits)	152.92 ± 56.23	150.82 ± 56.11
Experiment	$N_R=7, D_F=5, M_T=0$	$N_R=10, D_F=5, M_T=0$
Vortex 256 (16 bits)	151.43 ± 56.55	162.26 ± 59.04
SHA 256 (16 bits)	149.55 ± 55.88	154.34 ± 56.71
Experiment	$N_R=3, D_F=5, M_T=1$	$N_R=5, D_F=5, M_T=1$
Vortex 256 (16 bits)	150.31 ± 56.55	149.14 ± 55.41
SHA 256 (16 bits)	150.45 ± 57.09	147.32 ± 55.15
Experiment	$N_R=7, D_F=5, M_T=1$	$N_R=10, D_F=5, M_T=1$
Vortex 256 (16 bits)	154.68 ± 56.42	153.66 ± 57.21
SHA 256 (16 bits)	151.31 ± 56.74	150.29 ± 57.98