Ｈｙ Ｒ Ａ Ｌ

(Hybrid Randomization Algorithm)

自 己 評 価 書

㈱ローレルインテリジェントシステムズ

平田　耕蔵

J-ver2

目　　次

１．設計思想

ＨｙＲＡＬを設計するにあたって留意した事は現在知られている解読技術に充分な耐性を得ることが前提であるが、この暗号技術を使用して以下の３つの使用方法に発展できる事を意図したものである。

①１つのキー１回のブロック暗号の計算で暗号と認証の両方の目的を達成できる事。

これは通信またはファイル等においては秘匿性と保全性の両方が要求される。この両方の目的を達成する為には、現在推奨される方法ではブロック暗号の２回分に相当する計算が要求されている。パフォーマンス上から考えて１回の計算である方が望ましい。

②ブロック暗号技術を利用したハッシュ関数の適用。

ハッシュ関数については、現在米国においてＳＨＡ－３の選定が行われ近々標準的なものとして選定される予定である。これまで広く知られているハッシュ関数は、ブロック暗号とは異なる独自のアルゴリズムであり、ブロック暗号に比べてパフォーマンスにおいて優位であったが、解読技術の進歩によりこれまでのものは充分なる安全性が得られなくなった事から、ＳＨＡ－３の公募がなされた。公募のどのようなアルゴリズムがＳＨＡ－３として選定されるかは別としてパフォーマンスにおいて同等レベルであれば、ブロック暗号技術の応用において目的が達せられることが望ましいと考える。

③マルチブロック暗号

最近は Variable length のブロック暗号も要求として上がっている。固定したブロックサイズのブロック暗号の繰り返しではない方法による可変長のブロック暗号に対応できる事も設計思想として視点に入れたものである。

以上３点を考慮しつつ、そのプログラムにおいて一般的な簡単な手法をとすることにも注力した。プログラミングの殆どは、テーブル検索とワード（4byte）単位の Xor で可能としている。

２．設計思想に基づくＨｙＲＡＬの構成

２．１．s-box

ブロック暗号における非線形部分の最小単位として byte 単位の s-box がある。

ＨｙＲＡＬにおける s-box はＡＥＳのそれと同じような考え方で何ら独自性はない。

ただＡＥＳの場合はＳ層として 1/x 関数を使用し、その後でＰ層としての線形処理を行った結果を s-box としているが、ＨｙＲＡＬの s-box の場合はＰ層を先に行った後 1/x 関数を使用して作成している。

この違いがあるだけでＨｙＲＡＬの最大線形確率、最大差分確率は共に 2^{-6} である

２．２．ＭＤＳ変換

最小の非線形処理として byte 単位変換は s-box を通して行うが、これを 4byte 単位（32bit）で線形変換を行う為にＭＤＳ変換を行う。ＡＥＳにおける Mix column と同じ考え方である。

ＨｙＲＡＬではこのＭＤＳ変換にアンバランスなＭＤＳ変換の考え方を導入しているところに最大の特徴がある。

多くの暗号で使用されているＭＤＳ変換の殆どはバランス型である。これはＭＤＳに入力する 4byte の変数 x=($x1$、$x2$、$x3$、$x4$)において、$x1$= $x2$= $x3$= $x4$ が成り立つ時、x はバランスしていると呼び、入力がバランスしている時は出力もバランスする事を意味している。

ＨｙＲＡＬの場合は非巡回型のＭＤＳ行列を使用しているのでバランスしない。

ブロック暗号においては、多くの場合ＭＤＳに入力されるものは段キーと呼ばれるサブキーとのXor されたものがＳ層で変換される為、バランスがとれている事は直接弱点とはならない。

ブロック暗号の全体の中で多くのサブキーが使用され、そのすべてがバランスされたものでないからである。もし全ての段キーがバランスしているならばブロック暗号に入力する平文が全てバランスしているものを入力すれば、暗号文は必ずバランスされる事になる。

ブロック暗号においては上に述べた如く、バランスしている事が弱点とはならないかもしれないが、未知の攻撃法に結びつく可能性も考えてＨｙＲＡＬでは非巡回型のＭＤＳ行列を使用する事にした。

２．３．ＭＤＳ行列式

４×４の行列がＭＤＳ行列である事は、その任意の i×i 小行列式(i=1,2,3,4)が非零である事と等価である。

ＨｙＲＡＬの４×４ＭＤＳ行列は、発見的に以下の手順で求めた。

①正則な４×４行列の生成

②入力の 1byte のみを非零とした時、出力の 4byte が全て非零か検査し、不成立なら①へ。

③入力の 2byte のみを非零とした時、出力の 3byte が全て非零か検査した。
言い換えればその様な入力を適切に選ぶならば、1byte のみ零にできる。不成立なら①へ。

④入力の 3byte のみを非零とする入力を適切に選ぶならば、出力の 2byte まで零にできるか。
不成立なら①へ。

尚、各ステップで条件が不成立の場合、その不成立状態を回避する様に、元の正規行列に基本操作を加えた。この手順は視察により発見的に実行したので、誤りの確認を兼ねて最終的なＭＤＳ行列候補に対し、i×i 小行列式(i=1,2,3,4)の非零性検査を計算機で行い確認した。

２．４．基本関数（ f_i関数）

ＨｙＲＡＬでは２．２．項で述べた様なアンバランスなＭＤＳ行列を使用し、f_i関数として定義する。

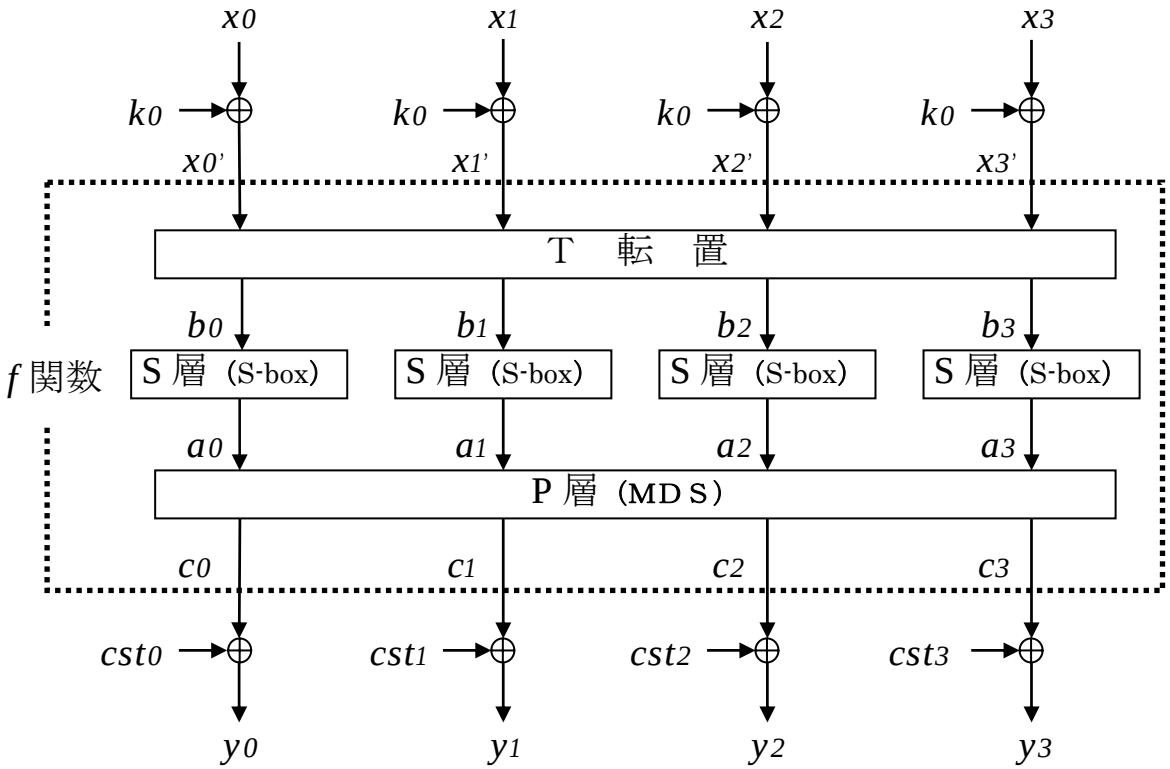


図１： f_i関数

アンバランスなＭＤＳ行列部分は次の２つの行列を合成したものである。

$$\begin{pmatrix} c0 \\ c1 \\ c2 \\ c3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \\ 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b0 \\ b1 \\ b2 \\ b3 \end{pmatrix} \qquad \begin{pmatrix} b0 \\ b1 \\ b2 \\ b3 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 2 & 1 \\ 2 & 4 & 1 & 1 \\ 4 & 5 & 2 & 1 \\ 1 & 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} a0 \\ a1 \\ a2 \\ a3 \end{pmatrix}$$

$$\begin{pmatrix} c0 \\ c1 \\ c2 \\ c3 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 \\ 7 & 3 & 1 & 2 \\ 7 & 4 & 5 & 3 \end{pmatrix} \begin{pmatrix} a0 \\ a1 \\ a2 \\ a3 \end{pmatrix}$$

図１における *CSTi* の加算は *a0*、*a1*、*a2*、*a3* が全て 0X00 の時の為である。

CSTi はテーブル内に含めるため検索は必要としない。

f_i関数は、テーブル検索４回とXor３回である。

バランスドなＭＤＳ行列を使用してもその出力に対して定数値（ＣＳＴ）を加算してバランスを崩すことは出来る。この様にしてバランスを崩している提案も見られた。しかしこの場合でも差分のバランスは崩す事は出来ない。

この基本的なＭＤＳ変換がバランスしていると、この機能をハッシュ関数に使用するには問題となる。

ＨｙＲＡＬのＭＤＳ変換がアンバランスドであるからf_i関数として（i=1～8）f関数を８つの種類にまで入力転置が生かされる事になる。

この事は設計思想で述べたブロック暗号をベースにして、他の暗号モードの拡張に発展する事が出来る一つの要素である。

２．５．拡大関数 **G1**、**G2**、**F1**、**F2**

f_i関数を利用した各拡大関数は４ラウンドで構成している。

G1、**G2** 関数は３入力１出力のFeistel型で、f_i関数は各ラウンドで１回の計算である。

F1、**F2** 関数は２入力１出力のFeistel型で、f_i関数は各ラウンドで２回の計算であり、縦接続を行ったものである。

この４つの関数は各々逆関数を持っていてそれ自身が 128bit のブロック暗号の機能を有している。

ＨｙＲＡＬでは４つの関数を簡単な縦接続して暗号化を図っており、鍵長が 128bit の場合は、**G1→F2→F2→F1→F1→G2** の構成をなしており、鍵長が 192、256bit の場合は **G1→F2→F2→F2→F1→F1→F1→G2** の構成をなしており、鍵長が 128bit の場合を Single Key Mode と呼び、鍵長が 192、256bit の場合を Double Key Mode と呼んでいる。

２分割の Feistel 型とは異なり、複合は各々の逆関数が使用され、暗号の順と逆順にサブキーを使用する。

２．６．全体フロー

Single Key Mode の全体のフローとラウンド鍵（**RK**）、中間鍵（**IK**）、の関係を次ページの**図２**に示す。

図２において各ラウンドにおける$\triangle Pi$は、f_i関数の出力を示している。$\triangle Pi$は、図に示した順序でXorがとられ１段毎に暗号化が進むようになっている。$\triangle Pi$の内、$\triangle P5$ から$\triangle P20$ まではf_i関数縦接続の出力結果である。各々のラウンドに使用するf_i関数はf_1からf_8まで使用される。

f_i関数の縦接続の２回目の処理前に入力される鍵が中間鍵（**IK**）で１ラウンドに 32bitずつ使用する。

f_i関数は 32bit入力 32bit出力の暗号ブロックでありPermutation Resultの意味を込めて$\triangle Pi$と表記する事にした。

今回のブロック暗号では$\triangle Pi$ として表記するに留まっているがＨｙＲＡＬブロック暗号の拡張モードとして使用する場合には内部連鎖情報としてこの$\triangle Pi$ が大きな意味を持つことになる。

ＨｙＲＡＬブロック暗号は 8bit 単位の最小の Non-Linear s-box を使用し、32bit 単位の８種類ブロック暗号を構成し、この 32bit 単位のブロック暗号を 128bit のブロック暗号へと拡大した４種類の128bit ブロック暗号を使用して暗号化を図っていると言える。

そして最大の特長は鍵に依存しない Permutation 構造をとっている事である。

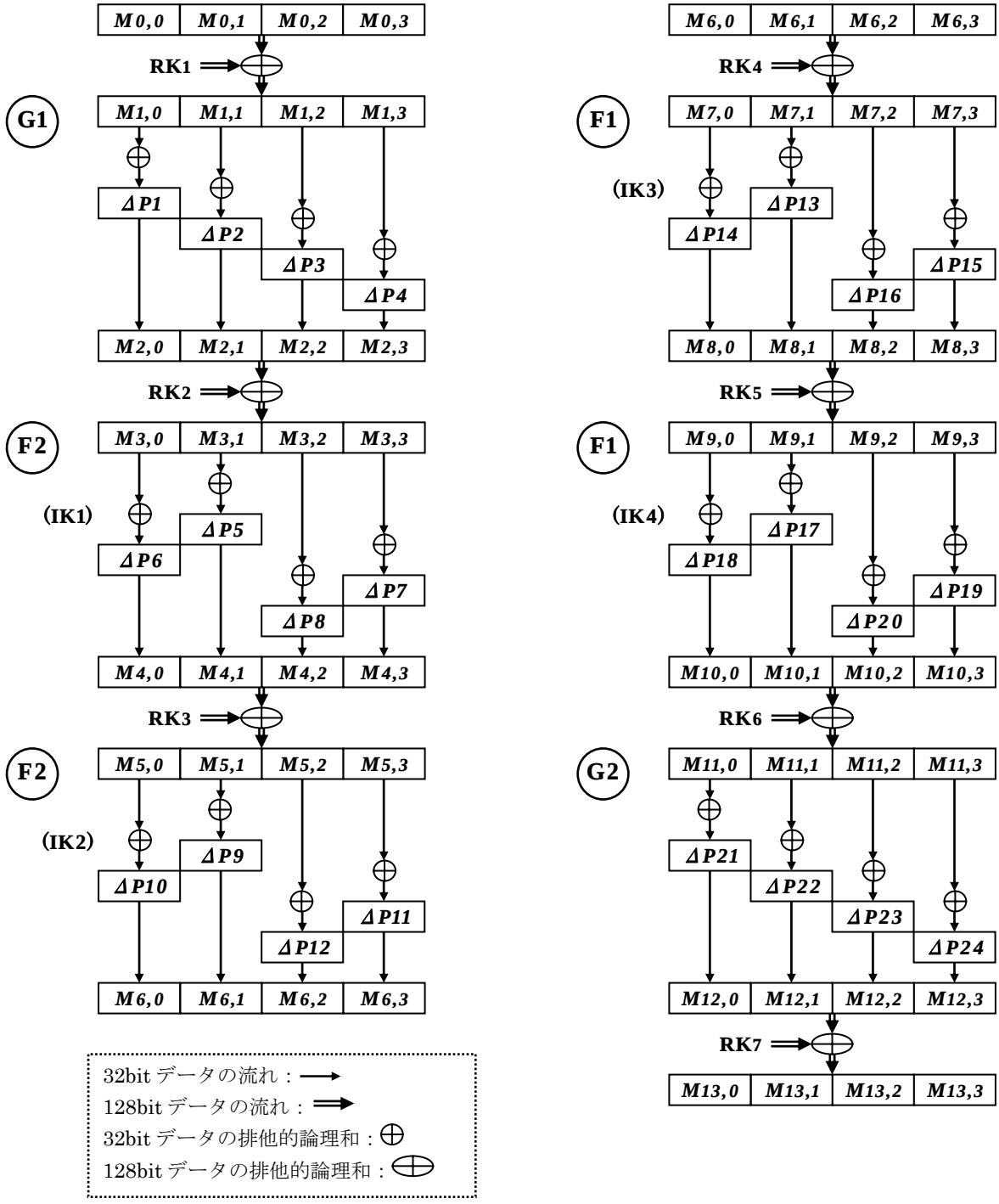

M0,0 M0,1 M0,2 M0,3
RK1
G1
M1,0 M1,1 M1,2 M1,3
ΔP1
ΔP2
ΔP3
ΔP4
M2,0 M2,1 M2,2 M2,3
RK2
F2
M3,0 M3,1 M3,2 M3,3
(IK1)
ΔP5
ΔP6
ΔP7
ΔP8
M4,0 M4,1 M4,2 M4,3
RK3
F2
M5,0 M5,1 M5,2 M5,3
(IK2)
ΔP9
ΔP10
ΔP11
ΔP12
M6,0 M6,1 M6,2 M6,3
M6,0 M6,1 M6,2 M6,3
RK4
F1
M7,0 M7,1 M7,2 M7,3
(IK3)
ΔP13
ΔP14
ΔP15
ΔP16
M8,0 M8,1 M8,2 M8,3
RK5
F1
M9,0 M9,1 M9,2 M9,3
(IK4)
ΔP17
ΔP18
ΔP19
ΔP20
M10,0 M10,1 M10,2 M10,3
RK6
G2
M11,0 M11,1 M11,2 M11,3
ΔP21
ΔP22
ΔP23
ΔP24
M12,0 M12,1 M12,2 M12,3
RK7
M13,0 M13,1 M13,2 M13,3
32bit データの流れ：
128bit データの流れ：
32bit データの排他的論理和：
128bit データの排他的論理和：

図２

３．ベースとして用いる理論・技術

s-boxの設計に対しては、現在 8bitのs-boxとして、差分および線形の最大確率 2^{-6}を確保できる。

x^{-1}関数（MOD ^8+x^4+x^3+x+1）をベースに線形変換の後、このx^{-1}関数を使用して作成したものである。線形層を通してもx^{-1}関数の差分および線形の確率はそのまま継承し、不動点および対極点を無くすことが出来ている。

s-box を Non-Linear Permutation とし、これを 32bit 単位で取り扱う場合、最小の分岐数を“5”と言う事を実現する為にＭＤＳ変換がある。これまで多くの暗号技術の中で、ＭＤＳ変換を使用していたが、ＡＥＳと同様に全てが balanced 型と言える。

balanced 型なＭＤＳ変換ではキーのみによって balance が崩されている。

ＨｙＲＡＬでは un-balanced なＭＤＳ変換を使用した。

最終的に使用したＭＤＳ変換が正しいかどうかの検証を行い決定した。検証方法は小行列 i×i 小行列式(i=1,2,3,4)のひ非零性検査を計算機で行い確認した。

ＭＤＳ変換における分岐数“5”を確保し、且つ un-balanced なＭＤＳ変換を開発、使用した。

４．安全性に対する評価

先に発表したＨｙＲＡＬ[1]では差分確率および線形確率[2]において安全性を示す上界が得られなかった。そこでＨｙＲＡＬの安全性を得る為の改良を行った。

また、差分および線形の耐性についてより厳密な評価を行うべく一部専門家に依頼した。その結果、線形および差分の特性確率の上界は、秘密鍵 128bitで各々2^{-228}、2^{-222}であり、秘密鍵 192bit、256 bit でそれぞれ 2^{-318}、2^{-342}を示す事が出来た。

ＳＣＩＳ２０１０暗号と情報セキュリティシンポジウム（2010 年１月）において東京理科大学金子教室より、8bitトランケーションによる線形 [3] および差分 [4] の最大特性確率の上界は、前述のように発表された。今後、引き続き他の攻撃に対する耐性についても.発表していく予定である。

［1］平田耕蔵、共通鍵暗号 HyRAL16、信学技報　ISEC　2006-76（2006-09）P29

［2］五十嵐保隆・金子敏信・福林直人、共通鍵ブロック暗号 HyRAL の線形攻撃耐性評価、信学技報　ISEC　2006-157（2007-03）P99

［3］五十嵐、高木、金子：共通鍵ブロック暗号 HyRAL の線形攻撃耐性評価、SCIS2010, 1D1-3

［4］高木、五十嵐、金子：共通鍵ブロック暗号 HyRAL の差分攻撃耐性評価、SCIS2010, 1D1-2

５．ソフトウェアの実装性評価

５．１．リソース使用量

コード量：104KB[ソース:96KB、ヘッダー：8KB]

ワークエリア：最大:4,888byte[最小:4,816byte]

※ワークエリアのサイズはコンパイルオプションにより変化する為、最大値と最小値を示す

５．２．記述言語

ANSI C

５．３．評価方法

項番	評価内容
5.3.1	エンディアン評価
5.3.2	複数プラットフォーム評価
5.3.3	速度評価
5.3.4	AES との比較評価

５．３．１．エンディアン評価

リトルエンディアン、ビッグエンディアンの両実機で評価を行いました。

ツール「HyRAL 暗号_20091214.xls」および「HyRAl 復号_20091214.xls」と同一計算結果で在る事を確認。

評価を実施したマシンおよび OS は下記に示します。

Endian	CPU	Memory	OS
リトル	Intel CoreDuo 1.8GHz	512MB	Mac OS X 10.5 Leopard
ビッグ	PowerPC G4 867MHz[デュアル]	768MB	Mac OS X 10.4 Tiger

※コンパイル・オプションでエンディアン指定する事により、各エンディアンに対応する事が可能です。

５．３．２．複数プラットフォーム評価

複数プラットフォームでの動作を評価しました。

ツール「HyRAL 暗号_20091214.xls」および「HyRAl 復号_20091214.xls」と同一計算結果で在る事を確認。

評価を実施したマシンおよび OS は下記に示します。

Endian	CPU	Memory	OS
リトル	Intel Core2 Duo 2.33GHz	2GB	Windows XP Pro SP3
リトル	AMD Athlon64 2800+ 1.8GHz	1GB	Windows XP Pro SP3
リトル	Intel Core Duo 1.8GHz	512MB	Mac OS X 10.5 Leopard
ビッグ	PowerPC G4 867MHz[デュアル]	768MB	Mac OS X 10.4 Tiger

５．３．３．速度評価

・速度評価方法

１．SetThreadAffinityMask を使用して、指定したスレッドを実行可能なプロセッサを指定（限定）します（デュアル CPU 対応）

２．アセンブラ命令 rdtsc でクロックを取得(c0)

３．測定する処理（鍵スケジューリング／暗号化処理／復号化処理）を一定回数繰り返す

４．アセンブラ命令 rdtsc でクロックを取得(c1)

５．clock = c1 - c0 で処理にかかったクロックを計測

６．上記処理を 10 ケース繰り返し clock の最小値を出力

実測 Hz	評価マシンで計測した CPU の Hz
Clock	c1 － c0
Loop	評価に辺り繰り返し処理した回数
Clock/1Block(1keyGen)	1Block にかかる ClockCycles を計測：Clock / Loop
Sec/1Block(1KeyGen)	1Block にかかる秒数を計測：(Clock / Loop) / 実測 Hz

・速度評価方法

評価を実施したマシンおよび OS は下記に示します。

Endian	CPU	Memory	OS
リトル	Intel Core2 Duo 2.33GHz	2GB	Windows XP Pro SP3
リトル	AMD Athlon64 2800+ 1.8GHz	1GB	Windows XP Pro SP3

・速度評価結果

<<Key 生成[128bit]>>

CPU	実測 Hz	Clock	Clock/1KeyGen	Sec/1keyGen	Loop
Intel Core2 Duo 2.33GHz	2346708812	614634370	540.491574000	0.000000230	1000000
AMD Athlon64 2800+ 1.8GHz	1844040188	536723556	533.621363000	0.000000289	1000000

<<暗号化処理[128bit]>>

CPU	実測 Hz	Clock	Clock/1Block	Sec/1Block	Loop
Intel Core2 Duo 2.33GHz	2346708812	676443936	668.277442000	0.000000284	1000000
AMD Athlon64 2800+ 1.8GHz	1844040188	659460428	656.071862000	0.000000356	1000000

<<復号化処理[128bit]>>

CPU	実測 Hz	Clock	Clock/1Block	Sec/1Block	Loop
Intel Core2 Duo 2.33GHz	2346708812	662740414	662.740414000	0.000000282	1000000
AMD Athlon64 2800+ 1.8GHz	1844040188	654356835	653.824521000	0.000000355	1000000

<<Key 生成[256bit]>>

CPU	実測 Hz	Clock	Clock/1KeyGen	Sec/1keyGen	Loop
Intel Core2 Duo 2.33GHz	2346708812	1337548856	1212.252034000	0.000000515	1000000
AMD Athlon64 2800+ 1.8GHz	1844040188	1153811271	1146.935333000	0.000000622	1000000

<<暗号化処理[256bit]>>

CPU	実測 Hz	Clock	Clock/1Block	Sec/1Block	Loop
Intel Core2 Duo 2.33GHz	2346708812	891542778	891.542778000	0.000000379	1000000
AMD Athlon64 2800+ 1.8GHz	1844040188	887742704	886.610207000	0.000000481	1000000

<<復号化処理[256bit]>>

CPU	実測 Hz	Clock	Clock/1Block	Sec/1Block	Loop
Intel Core2 Duo 2.33GHz	2346708812	908677224	898.772084000	0.000000382	1000000
AMD Athlon64 2800+ 1.8GHz	1844040188	905184049	900.269290000	0.000000488	1000000

５．３．４．AES との比較評価

・比較評価方法

「５．３．３．速度評価」と同様の速度計測を AES で実施し、HyRAL と比較しました。

ただし、計測した AES はアセンブラを含むソースを利用している為、正確な比較は出来ておりません。

また、AES との比較は暗号化処理のみ実施しました。

・比較評価方法

評価を実施したマシンおよび OS は下記に示します。

Endian	CPU	Memory	OS
リトル	Intel Core2 Duo 2.33GHz	2GB	Windows XP Pro SP3
リトル	AMD Athlon64 2800+ 1.8GHz	1GB	Windows XP Pro SP3

・比較結果

<<暗号化処理[128bit]>>

Algorithm	CPU	実測 Hz	Clock	Clock/1Block	Sec/1Block	Loop
HyRAL	Intel Core2 Duo 2.33GHz	2346708812	676443936	668.277442000	0.000000284	1000000
	AMD Athlon64 2800+ 1.8GHz	1844040188	659460428	656.071862000	0.000000356	1000000
AES	Intel Core2 Duo 2.33GHz	2346708812	327552806	323.409156000	0.000000138	1000000
	AMD Athlon64 2800+ 1.8GHz	1844040188	512242619	511.463520000	0.000000276	1000000

128bit 暗号処理「Intel CPU」では、HyRAL に比べ AES が約 51%速い

128bit 暗号処理「AMD CPU」では、HyRAL に比べ AES が約 22%速い

<<暗号化処理[256bit]>>

Algorithm	CPU	実測 Hz	Clock	Clock/1Block	Sec/1Block	Loop
HyRAL	Intel Core2 Duo 2.33GHz	2346708812	891542778	891.542778000	0.000000379	1000000
	AMD Athlon64 2800+ 1.8GHz	1844040188	887742704	886.610207000	0.000000481	1000000
AES	Intel Core2 Duo 2.33GHz	2346708812	489961458	459.049990000	0.000000196	1000000
	AMD Athlon64 2800+ 1.8GHz	1844040188	696486615	696.486615000	0.000000377	1000000

６．ハードウェアの実装性評価

ＨｙＲＡＬＦＰＧＡ設計仕様書に基づき設計されたＦＰＧＡの評価です。

６．１．評価環境

６．１．１．ＦＰＧＡ

ターゲットとしたＦＰＧＡは､Xilinx 社製 XC5VLX30 である。

６．１．２．シミュレーション環境

シミュレーターツールは、MentorGraphlcs 社製 Model-Slm を使用した。

６．１．３．論理合成-配置配線環境

論理合成および配置配線ツールは、Xilinx 社製 ISE を使用した。

６．２．評価結果

評価は Model-Sim による論理的検証と ISE による論理合成、配置配線結果にて行う。

６．２．１．Model-Sim による論理的検証

Model-Sim を用いたハードウェアの検証結果とソフトウェアにより求めた期待値の照合結果一覧表は６．２．３．を参照。。

シミュレーションバターンは、ローレルバンクマシン株式会社支給のパターンにて実施した。

シングルキーモードおよびダブルキーモードにおいてそれぞれ１０パターンのキーに対し、１２８パターンの平文を暗号化し、確認した。

キーおよび平文のテストデータはテストベクトルに同じ。

６．２．２．処理時間

処理時間は、20MHz クロック(50ns)で動作させた場合の時間である。

・Sub Key 生成時間

BSTART-KEYEND(SinglaKayMode):6.3μs(126CLK)

BSTART-KEYEND(DoubleKeyMode):12.55μs(251 CLK)

・暗号文生成時間

CSTART-EEND(SinglaKayMode):11.5μs(230CLK)

CSTART-EEND(DoubleKayMode)16.0μs(320CLK)

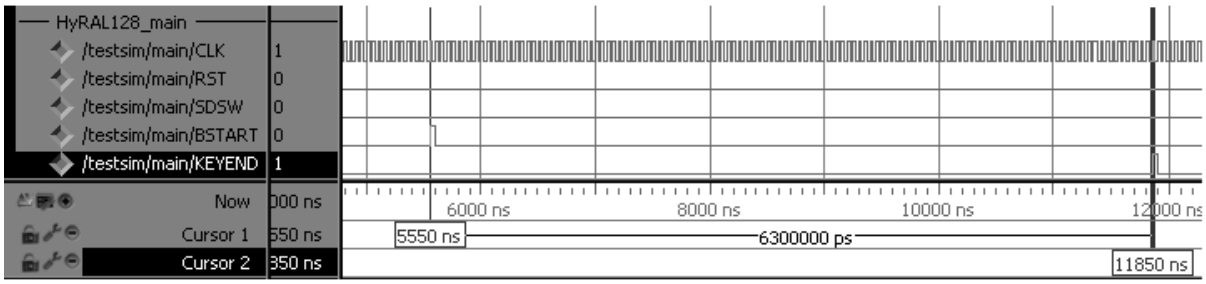
・復号文生成時間

DSTART-DEND(SlngleKeyMode)11.55μs(231CLK)

DSTART-DEND(DoubleKayMode)16.05μs(321CLK)

・SubKey 生成時間 BSTART～KEYEND まで

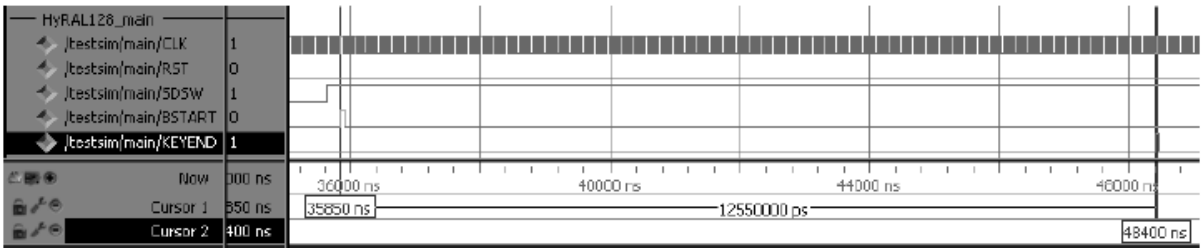
BSTART-KEYGEN(SlngleKeyMode):6.3μs



BSTART はアドレス 0x0 の 0bit 目に定義される Key 生成開始信号です。

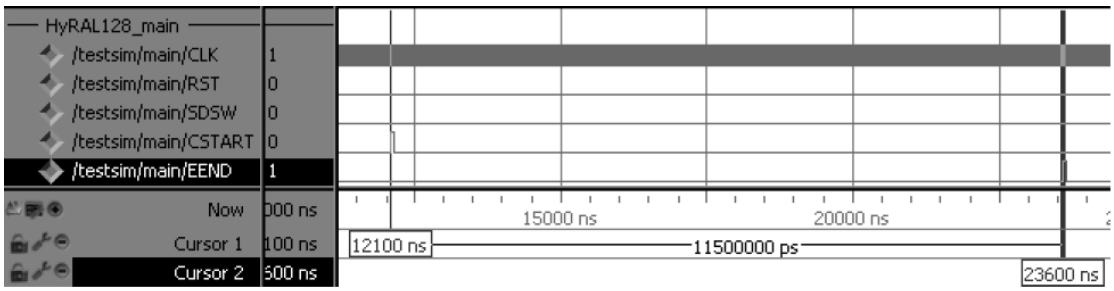
KEYEND は SUBKEY の生成終了を意味する信号です。

BSTART-KEYGEN(DoubleKayMode):12.55μs

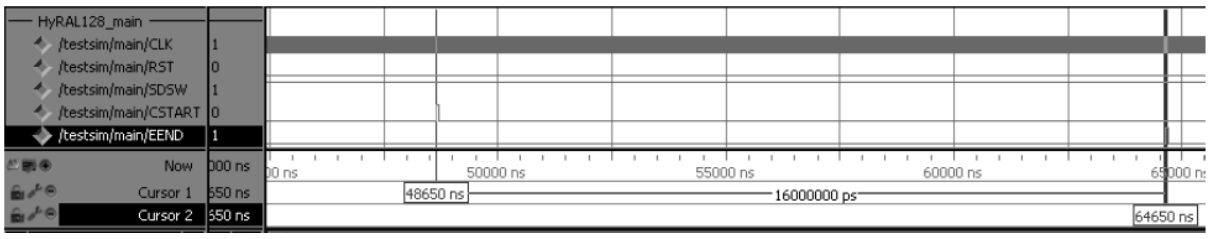


・暗号文生成時間 CSTRAT～EEND まで

CSTART-EEND(SlngleKeyMode):11.5μs

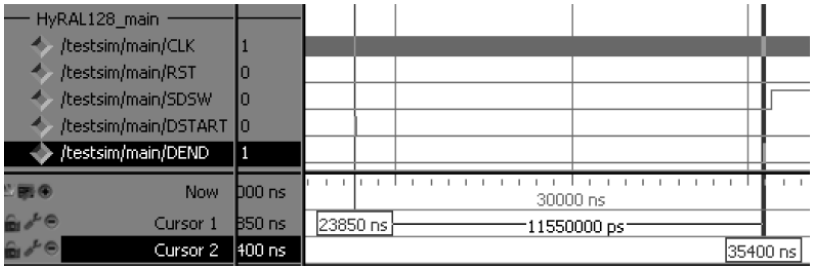


CSTART-EEND(DoubleKayMode):16.0μs

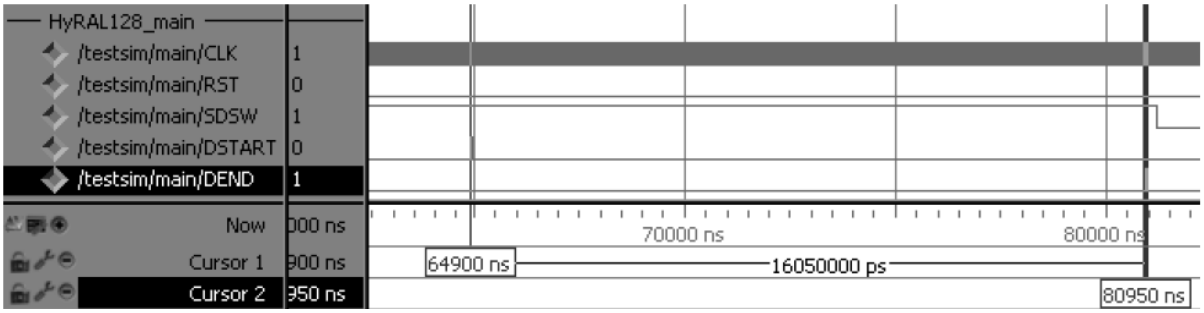


・復号文生成時間 DSTRAT～DEND

DSTART-DEND(SlngleKeyMode):11.55μs



DSTART-DEND(DoubleKayMode):16.05μs



6．2．3．Xilinx 社製 ISE による論理合成・配置配線結果

・Optimization = Speed

HyRAL_ALL Project Status(01/01/2010-06:53:10)			
Project File:	HyRAL_ALL..ise	**Carrent State:**	Placed and Routed
Module Name:	HyRAL128_main	**Errors:**	No Errors
Target Device:	xc5vlx30-3ff324	**Warnings:**	5 Warnings
ProdactVersion:	ISE10.1-WebPACK	**Routing Results:**	All Signals Completely Routed
Design Goal:	Balanced	**Timing Constraints:**	All Constraints Met
Design Strategy:	Xilin x Default(unlocked)	**Final Timings Score:**	0(Timing Report)

HyRAL_ALL Partition Summary
No partition in formation was found.

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	7,271	19,200	37%	
Number used as Flip Flops	7,271			
Number of Slice LUTs	9,211	19,200	47%	
Number used as logic	9,211	19,200	47%	
Number using O6 output only	9,211			
Slice Logic Distribution				
Number of occupied Slices	3,135	4,800	65%	
Number of LUT Flip Flop pairs used	11,025			
Number with an unused Flip Flop	3,754	11,025	34%	
Number with an unused ULT	1,814	11,025	16%	
Number of fully used LUT-FF pairs	5,457	11,025	49%	
Number of unique control sets	53			
IO Utilization				
Number ofbonded IOBs	71	220	32%	
Specific Feature Utilization				
Number of Block RAM/FIFO	2	32	6%	
Number using Block RAM only	2			
Total primitives used				
Numer of 18k Block RAM used	4			
Total Memory used(KB)	72	1,152	6%	
Numer of BUFG/BUFGCTRLs	2	32	6%	
Number used as BUFGs	2			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout report
Routing Results:	All Signals Comple tely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	±1 2 04:09:03 2010	0	4 Warnings	46 Infos
Translation Report	Current	±1 2 06:35:41 2010	0	1 Warning	0
Map Report	Current	±1 2 06:47:37 2010	0	0	7 Infos
Place and Route Report	Current	±1 2 06:51:48 2010	0	0	0
Static Timing Report	Current	±1 2 06:53:09 2010	0	0	2 Infos
Bitgen Report					

・Optimization = Area

HyRAL_ALL Project Status(01/01/2010-12:01:50)			
Project File:	HyRAL_ALL..ise	**Carrent State:**	Placed and Routed
Module Name:	HyRAL128_main	**Errors:**	No Errors
Target Device:	xc5vlx30-3ff324	**Warnings:**	751 Warnings
ProdactVersion:	ISE10.1-WebPACK	**Routing Results:**	All Signals Completely Routed
Design Goal:	Area Reduction	**Timing Constraints:**	All Constraints Met
Design Strategy:	strategy1	**Final Timings Score:**	0(Timing Report)

HyRAL_ALL Partition Summary
No partition in formation was found.

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	7,239	19,200	37%	
Number used as Flip Flops	7,239			
Number of Slice LUTs	8,576	19,200	44%	
Number used as logic	8,576	19,200	44%	
Number using O6 output only	7,804			
Number using O5 output only	6			
Number using O5 end O6	766			
Slice Logic Distribution				
Number of occupied Slices	3,245	4,800	67%	
Number of LUT Flip Flop pairs used	10,171			
Number with an unused Flip Flop	2,932	10,171	28%	
Number with an unused ULT	1,595	10,171	15%	
Number of fully used LUT-FF pairs	5,644	10,171	55%	
Number of unique control sets	52			
IO Utilization				
Number ofbonded IOBs	71	220	32%	
IOB Flip Flops	67			
Specific Feature Utilization				
Number of Block RAM/FIFO	2	32	6%	
Number using Block RAM only	2			
Total primitives used				
Numer of 18k Block RAM used	4			
Total Memory used(KB)	72	1,152	6%	
Numer of BUFG/BUFGCTRLs	2	32	6%	
Number used as BUFGs	2			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout report
Routing Results:	All Signals Comple tely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

６．２．４．Synthesis Report

以下に ISE が生成する Synthesis Report 内の 4Warning について内容確認を行った。

```
---------------------------------------------------------------------------------
WARNING:Xst2211 -"sbox_rom.v" line 108: Instantiating black box module<sbox_rom>.
WARNING:Xst2211 -"sbox_rom.v" line 113: Instantiating black box module<sbox_rom>.
WARNING:Xst2211 -"sbox_rom.v" line 118: Instantiating black box module<sbox_rom>.
WARNING:Xst2211 -"sbox_rom.v" line 123: Instantiating black box module<sbox_rom>.
---------------------------------------------------------------------------------
```

上記ワーニングは、CoreGenerator で生成した sbox_rom というモジュールをブラックボックスで使用している旨のワーニングであり、問題ない。

６．２．５．Translation Report

以下に ISE が生成する Translation Report 内の 1Warning について内容確認を行った。

```
---------------------------------------------------------------------------------
WARNING:NgdBuild:478 -clock net RST_IBUF with clock driver RST_IBUF_BUFG drives no clock pins
---------------------------------------------------------------------------------
```

上記ワーニングは、RST 信号が DCM によって生成されたが、フリップフロップなどのクロックピンをドライブしていないワーニングであることを示している。これは、非同期リセットを回路で使用しているためであり、問題ない。

６．２．６．Static Timing Report

以下に ISE が生成する Static Timing Report 内の内容確認を行った。

・Optimization = Speed

```
=================================================================================
Timing constraint: TS_CLK=PERIODTIMEGRP "CLKI" 20nsHIGH50%:

 100886 paths analyzed, 27772 endpoints analyzed, 0 failing endpoints
 0 timing errors detected. (0 setup errors, 0 holderrors)
 Minimum period is     9.885ns.
---------------------------------------------------------------------------------
```

以上から、最終的な配置におけるワースト動作周波数は、101MHz(周期:9.885ns)という結果になった。

・Optimization = Area

```
=================================================================================
Timing constraint: TS_CLK=PERIODTIMEGRP "CLKI" 20nsHIGH50%:
 112569 paths analyzed, 25823 endpoints analyzed, 0 failing endpoints
 0 timing errors detected. (0 setup errors, 0 hold errors)
 Minimum periods       9.055ns.
---------------------------------------------------------------------------------
```

以上から、最終的な配置におけるワースト動作周波数は、110MHz(周期:9.055ns)という結果になった。

６．３．結論

HyRAL 略号化および復号化を FPGA に実装した結果６．２．１．により論理回路動作について、シミュレーションによる動作確認を行い、略号化および復号化が可能であることを確認した。シングルキーモード、ダブルキーモードで各 1280 パターンでの動作確認を行い、論理回路の正当性の確認もとれた。

また、処理時間については、６．２．１．に示すとおり最大 572CLK(=251CLK[KEY 生成]+321 CLK[復号])である。

論理合成および配置配線の結果、最大動作周波数は 110MHz であるため、全ての処理が終わるまでに 5.179μs 程度時間を要する結果となる。

実装面積については、Optimization による差はほぼ無い結果となっており、Xilinx 社製 XC5VLX30 の使用スライス数で 65%程度の使用率となったが、128、192、256bit の鍵が使用可能で、暗号および復号機能も盛り込んだ結果である。

また、ＦＰＧＡのＩＯピン数の制約があったため簡単なインターフェース回路も実装した状態での結果である。

インターフェース回路では、最低 1000 個程度のフリップフロップと付随する論理回路を使用しており、暗号・復号化で使用しているフリップフロップ数は 6200 個程度となる。

７．サイドチャネル攻撃について

７．１．キャッシュアタックに対して

ＨｙＲＡＬはＡＥＳと同様にテーブル参照を多く行っている。

その為、キャッシュ攻撃については、耐性を持たせる事には大きなコストを払わなければならないことになる。

キャシュ攻撃による最も直接的な攻撃は、暗号プログラムの実行中に同一プロセッサーにおいて監視するプログラムが、キャッシュメモリーを直接コントロールする事によりターゲットの暗号プログラムのキャッシュメモリーをどのように使ったかを知る事により、サブキーを順次解析していく方法がある。

大きなテーブル参照を行う、ブロック暗号はＨｙＲＡＬのみならず、全てこの危険性を持つものである。その為、テーブル参照を使用しない非線形処理が希望されるが、事前生成されるテーブルと比べて、同等の効果を得ることは難しい事と、他の攻撃との耐性との比較において、甲乙付けがたい点がある。

ＨｙＲＡＬでは、テーブル参照を行いつつキャッシュ攻撃に対して対抗策を検討する。

その１つは使用する４つのテーブルを１つのテーブルとし、テーブルに対するアクセスを分散させない方法を検討する。しかし、この方法では効率性において著しく低下させる事になる。

もう１つは Buzz 処理を行わせることである。これは最初のラウンドから数ラウンド実行させる。

Buzz 処理は全然無駄なプロセスでありこれもまた、効率性を低下させる要素がある。

キャッシュ攻撃のようなアルゴリズムに対する攻撃でないものに対しては、セキュリティレベルによりこれに対抗策をもったものを使用するのかどうかの選択は出来る様に考えておく。

それから、以下に述べる点は直接、このテーマとは関係が無いが今後のテーマとなるかと思い述べておく。

これまで共通鍵暗号の暗号モードには全て暗号の鍵は静的に使用する事が前提となっている。ＣＢＣ、ＣＦＢ、ＯＦＢもＥＣＢが基となっている。従って、攻撃者は常に鍵の状態が同じものに対して有利な条件で攻撃できる。キャッシュ攻撃などもその１つである。

もし、鍵が時刻と共に変わるならば、これまでの攻撃方法の統計手法は効をなさない。ＤＥＳが発表された当時から、Variable Key の方法はあったが、ハードウェアに馴染まないところから、暗号モードとしては採択されていない。しかし、実際のアプリケーションから考えて、ＥＣＢ単独でしようするのは、鍵暗号化オペレーションぐらいで、通信データとかファイルとか一連のデータの秘匿性および保全性の必要性は大きい。

今日、ハッシュ関数において様々な提案がなされているが、最近では内部情報連鎖手法が多くなってきている。レポートではステート情報とか Change Value とういう言葉で使われている。

ブロック暗号においてこの手法を用いればキャッシュ攻撃は効をなさないであろうと考える。

ＨｙＲＡＬでは次のテーマとしての、内部情報連鎖手法を意識した設計思想を持たせる方法を採り、これまでの鍵の静的な使用から動的なものへと応用していく予定である。

鍵を静的に使用するＥＣＢモードはマスター鍵による鍵暗号化オペレーションのようなもので、マスター鍵等は耐改ざん装置に封入する事を意識している為、キャッシュアタックが出来る様な環境下に置くべきでないという考え方である。

このような考え方であるため、ＥＣＢモードが使われるケースが限られる事を想定し、キャッシュアタックに対するオーバーヘッドが大きくてもシステム全体のパフォーマンスに大きな影響を与えないという前提で、安全性を重視した対応を検討し、発表していく予定である。

７．２．ＳＰＡ・ＤＰＡに対して

ハードウェアロジックが消費する電力および電圧の変化などを測定する事により秘密鍵の値を得ようとする攻撃に対しては、ハードウェアで出来る限り測定が意味をなさないような、平滑化を行わせるような実装を行う。ガイドラインに示されるような手段を取り入れる予定である。物理的な対策と共に内蔵させるソフトウェアに対しては、１つのアイデアとして次の様な事が考えられる。

これらの攻撃のもととなっているのは、情報（鍵およびデータ）が特にハミングウェイトに対するものである。もし、このハミングウェイトを平滑化すれば、少なくとも攻撃者としては測定の意味が無くなるであろうと考える。

その他に、ロジックを二つ組込む必要がある、１つは正規の計算であり、もう１つは平滑化するためのものダミーの計算である。

秘密キーとデータの排他的論理和をとり、テーブルを検索するのが大方の方式ではあるが、入力データは二つのプロセスを通る事にする。

１つは秘密キーとのＸＯＲがとられ正規なプロセスを図るが、もう１つは秘密キーのＮＯＴ値とＸＯＲが取られ、同様にプロセスを通る。電気使用量は２倍になるが、電気的な特性の測定には対抗策となる。これをアルゴリズム全体に行うかどうかは検討の必要がある。

対改ざん装置のようなものとは、装置が開けられ内部のチップを直接測定するようなことは出来ない。

装置が開けられたならば秘密鍵は揮発するように考えるのが通常である。

しかしＩＣカードのようなものは、他の装置からの電力の供給を受ける必要があるので、ロジックにおける対策が必要である。