Cryptanalysis

Lecture 11: Boolean Functions and Cryptanalysis

John Manferdelli
jmanfer@microsoft.com
JohnManferdelli@hotmail.com

Schedule

Date	Topic/Speaker
Jan 5, 2009	John: Error Correcting Codes and McEliece's Public Key System
Jan 12, 2009	John: Boolean Functions
Jan 19, 2009	No class, MLK day.
Jan 26, 2009	MOV attack. Dustin Moody. Guest lecture.
Feb 2, 2009	Linear and differential cryptanalysis of DES. Slava Chernyak, Sourav Sen Gupta.
Feb 9, 2009	Algebraic attacks, Paul Carr. MOV computation, Dan Shumow,
Feb 16, 2009	No class.
Feb 23, 2009	No class (Hash workshop in Cologne).
March 2, 2009	Attacks on MD4 and MD5. Owen Anderson. Factoring attacks. Wenhan Wang.
March 9, 2009	Attacks on stream ciphers. Karl Koscher (CSE).

Cryptanalytic Motivation

- Let E(**k, p**)= **c** be an enciphering operation and D(**k, c**)= **p** the corresponding deciphering operation with **k**$\in$GF(2)^k and **p, c**$\in$GF(2)^n. There are two canonical ways to "solve" the cryptanalytic problem for (E, D) under the chosen/corresponding plaintext attack:
  1. For fixed key, **k**, given corresponding plain and ciphertext pairs ($\mathbf{p_1}$, $\mathbf{c_1}$), ($\mathbf{p_2}$, $\mathbf{c_2}$), …, ($\mathbf{p_t}$, $\mathbf{c_t}$), find a function (program, procedure) which inverts E for an arbitrary ciphertext c. That is, find g, such that g(**c**)=**p**, if E(**k, p**)= **c** . ("Find the Inverse Function").
  2. Given corresponding plain and ciphertext pairs ($\mathbf{p_1}$, $\mathbf{c_1}$), ($\mathbf{p_2}$, $\mathbf{c_2}$), …, ($\mathbf{p_t}$, $\mathbf{c_t}$), find a function (program, procedure) that solves for **k**, that is, find h such that h(($\mathbf{p_1}$, $\mathbf{c_1}$), ($\mathbf{p_2}$, $\mathbf{c_2}$), …, ($\mathbf{p_t}$, $\mathbf{c_t}$))= **k**. In the simplest case, find h such that if E(h(**p,c**),**p**)-**c** =0. ("Find the Implicit Function").

- Either provides a "full service" break subject to the computational efficiency of finding and applying h and g respectively.

The Real World

- **Inverse Function Theorem:** Suppose $f : R^n \rightarrow R^n$ is continuously differentiable and $|\det(f'(a))| \neq 0$. $\exists V^{open}$, W^{open} and f^{-1}, such that $a \in V$, $f(a) \in W$ and $f^{-1}: W \rightarrow V$. Further, $f^{-1}(f(x)) = x$.
- **Implicit Function Theorem:** Suppose $f: R^n \times R^m \rightarrow R^n$ is continuously differentiable in an open set containing (a, b) and f(a, b) = 0 with $M = (D_{n+j}(f^i(a)))$ with $1 \leq i, j \leq m$. If $\det(M) \neq 0$, $\exists A^{open} \subseteq R^n$ and $B^{open} \subseteq R^m$ with $a \in A$, $b \in B$ such that $\forall x \in A$ there is a unique $g(x) \in B$ satisfying $f(x, g(x)) = 0$.

- Lesson: Differentiability and continuity make things simple in R.

Boolean Functions are different from real functions

1. The concept of differentiability is different (and less useful) in finite fields.
2. Things change “discontinuously” so the existence proofs for the inverse and implicit function theorems don’t carry over from the real case.
3. When inverses and implicit functions exist, they are not always easy to specify because they are not “continuous.”
4. All functions over finite fields can be represented as polynomials that is not true in the field of real numbers.
5. We can, in principle, construct a finite set containing every possible boolean function (for a fixed number of input and output variables), so we can in principle answer existence questions by exhaustive search of this list.
6. Constructing a “model” of how hard the inverse and implicit functions are to calculate is subtle in finite fields.

But wait…

- Not only “exact” solutions are useful. Even functions which meet the conditions of the implicit inverse function or implicit function theorem with high probability are quite valuable.

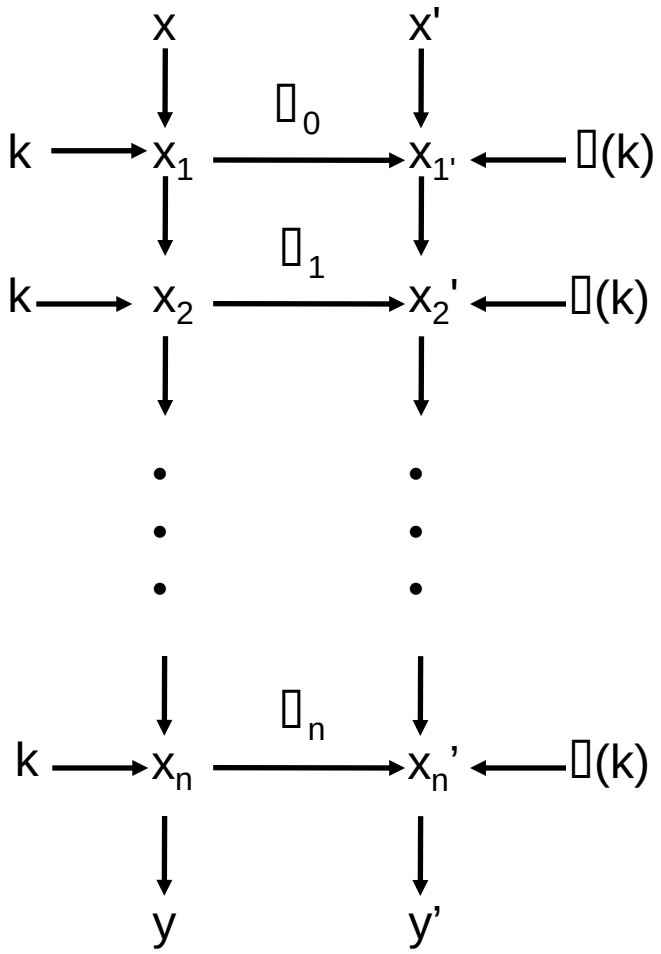
- Invariants or “constraints” of the form $a_1k_1 + \ldots + a_mk_m = f(\mathbf{p}, \mathbf{c})$ can help identify key bits even if they are right “slightly more frequently” than 1/2 . This represents a correlation between key bits and plain/cipher bits.

Simple examples of functional analysis

- We've used these ideas before opportunistically:
  - Linear solution to cipher systems.
  - Reduction to parallel systems with independent keys or plaintext segments.
  - Solving sparse equations over "linearized" variables to obtain inverses.
  - Using invariants to reduce key space searches in both linear and differential cryptanalysis.

- Now its time for a more systematic examination. This will allow us to completely determine inverse functions, implicit functions, correlations and invariant relationships.

Related Cryptosystems



- We can map an iterative cipher into a related cipher that is easier to solve.
- We did this in the case of “parallel” cryptosystems

Correlation coefficients

- Consider f: $GF(2)^n \rightarrow GF(2)$ and g: $GF(2)^n \rightarrow GF(2)$.
- Define C(f,g)= 2 Prob(f(x)=g(x))-1. C(f,g) describes the correlation between f and g.
- Now put $N=2^n$.
  - We can describe f as a vector in $GF(2)^N$ by setting **f**=(f(0,0,…,0), f(0,0,…,1),…, f(1,1,…,1)).
  - We can also embed **f** naturally in R^N: as follows:
  - $\mathbf{f_R} = ((-1)^{f(0)}, (-1)^{f(1)}, \ldots, (-1)^{f(N-1)})$.

Boolean Functions in Real Space

- Again let f:$GF(2)^n \rightarrow GF(2)$ and g: $GF(2)^n \rightarrow GF(2)$.
- Consider the two real vectors, in R^N , representing f and g. Define $\langle f,g\rangle = (\mathbf{f_R},\mathbf{g_R})$ and $||f|| = \sqrt{\langle f,f\rangle}$. With this notation, $C(f,g) = \langle f,g\rangle/(||f||\ ||g||)$.
- The vectors $(-1)^{\mathbf{w}} = (-1)^{\mathbf{w}\bullet\mathbf{x}}$ as **x** varies over $GF(2)^n$ are called the linear parities and form an orthogonal basis for R^N. Thus we can express any real function as a linear combination of the parities.

Boolean Functions and polynomials

- For Boolean f, V=GF(2)^m, f(v_1, v_2 , ... , v_m) = $\sum_{a \in V}$ g(a) v_1^{a1} v_2^{a2}... v_m^{am}
- g(a)= $S_{b \subseteq a}$ f(b1 , b2 , ... , bm) (subset means positions of 1's in a is a subset of b positions of 1's in b.)
- Theorem: If f is balanced, $\sum_w$ F(w) = $\pm 2^n$.
- Proof:

  $\sum_w$ F(w)= $\sum_w$ $\sum_x$ $(-1)^{f(x)+w \cdot x}$= $\sum_x$ $(-1)^{f(x)}$ ($\sum_w$ $(-1)^{w \cdot x}$)=

  $\sum_x$ $(-1)^{f(x)}$ 2^n $\delta_{w,x}$, so

  $\sum_x$ $(-1)^{w \cdot x+c}$ = $(-1)^c$ 2^n, w =0, 0, w≠0.

  Let F(w,c)= $\sum_x$ $(-1)^{f(x+w \cdot x+c)}$ then $\sum_{w,c}$ F(w,c)= 0.

Balance

- *Theorem:* If f:GF(2)$^{n-1}$ → GF(2) is any boolean function, g(x_1, ..., x_n)= f(x_1, ..., x_{n-1})+x_n is balanced.
- A balanced boolean function is uncorrelated with either constant function.
- Note that all balanced boolean functions can be obtained by applying a permutation in S_N to a sequence of N/2, 1's and N/2, 0's.
- If E_K: GF(2)^n → GF(2)^n, represents a block cipher, each component function must be balanced, that is have an equal number of 1 and 0 outputs in order to be invertible.

- *Generalized Balance Theorem:* For each $1 \le n \le 128$ and each $1 \le b_1 < b_2 < \ldots < b_n \le 128$ and fixed **k**, (E_{b1}(**k, x**), E_{b2}(**k, x**), ... , E_{bn}(**k, x**)) takes each value in GF(2)^n as **x** varies over GF(2)^n. So does any non-trivial sum of any of these functions.
- *Theorem:* A Boolean transformation is invertible iff every output parity is a balanced binary boolean function of the input bits.

Correlation matrices

- The correlation matrix, C, for a boolean function f, is a row matrix (indexed by w) defined by

  $C(f(x), w \bullet x) = \langle (-1)^{f(x)}, (-1)^{w \bullet x} \rangle$.

- A boolean *transformation* is a function **f**:$GF(2)^n \rightarrow GF(2)^m$. The definition of a correlation matrix can be extended to the vector valued boolean transformation **f** (consisting of m boolean functions) and, in this case, the correlation matrix, C, is a 2^m x 2^n matrix.
  - This matrix has entries $C_{uw} = C(u \cdot h(a), w \cdot a)$ where u indexes the rows and w indexes the columns; thus the u row can be represented as $(-1)^{u \bullet h(a)} = \sum_w C^{(h)}_{u,w} (-1)^{w \bullet a}$.
  - To emphasize the association with **f**, we sometimes write the correlation matrix as $C^{(\mathbf{f})}$.

Walsh transforms and correlation

- For boolean function, f: $GF(2)^n \rightarrow GF(2)$, define
  - $F(w)= 2^{-n} \sum_x (-1)^{f(x)+w \bullet x} = C(f(a),w \bullet a)$
  - We say $\mathcal{W}(f)=F$ and call $\mathcal{W}$ the normalized Walsh or Hadamard transform.
  - The term “Walsh Transform” is also used for the operation without the 2^{-n}, we will describe this as the “un-normalized” Walsh transform.
  - We’ve used Walsh transforms before to find the best affine approximators to boolean functions.

- Entries of the correlation matrix are Walsh transforms of component functions.

Walsh transforms: basic results

- Parseval: $\sum_w F(w)^2 = 1$.
- Convolution: $f \ast g(a) = \sum_x f(x)\, g(x+a)$.
- $\mathcal{W}^{-1}(F)(x) = f(x) = 2^{-n} \sum_t F(t)\, (-1)^{x \bullet t}$.
- $\mathcal{W}(f \ast g) = \mathcal{W}(f)\mathcal{W}(g)$.
- If f(x)= g(Mx+b), M, invertible, the absolute value of the spectrums of F and G are the same.
- dist(f(v), u•v)= ½ (2^n - 2^nF(u)).
- dist(f(v), u v+1)= 1/2(2^n + 2^nF(u)).
- $\mathcal{W}(f \oplus g) = \mathcal{W}(f) \otimes \mathcal{W}(g) = \sum_v F(v \oplus w)\, G(v)$.
- $\mathcal{W}(fg) = \frac{1}{2} (\delta(w) + \mathcal{W}(f) + \mathcal{W}(g) - \mathcal{W}(f \oplus g))$.

Fast Hadamard Transform

- Define $A \otimes B = (a_{ij} B)$.

- The operation is associative but not commutative.
- $N=2^m$, $I= 2^i$.
- $H_N = H_2 \otimes H_{N/2}$.
- $H_N = M^{(1)}_N M^{(2)}_N \ldots M^{(m)}_N$,
- $M^{(i)}_{N/2} I_{N/2} \otimes H_2 \otimes I_{IN/2}$.

Properties of component functions

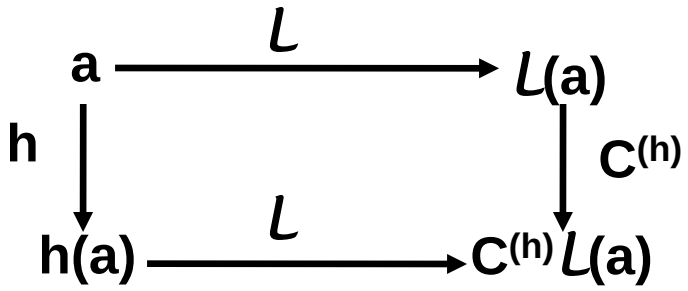
- Let f is a Boolean Function define S^0_f= {x: f(x)=0 } and S^1_f {x: f(x)=1}.
- If $e_i(x)= E_i(k,x)$ then $|S^b_{e1} \cap S^b_{e2} \cap \ldots \cap S^b_{ek}| = 2^{n-k}$.

- Note that all balanced boolean functions can be obtained by applying a permutation in S_N to a sequence of N/2, 1's and N/2, 0's.

- Counting Results: Let $N=2^n$ and BF(n) denotes the set of boolean functions on n-bit values then $|BF(n)|= 2^N$. $M=2^m$. Let BBF(n) be the balanced functions on n bits then $|BBF(n)|= {}_NC_{N/2}$, $|GA(n)| \sim 2^{M+m}$.

A correspondence

- The natural isomorphism L: $GF(2)^n \rightarrow R^N$ by $a \rightarrow (-1)^{a \cdot x}$.
- $L(a+b) = L(a)\ L(b)$ by pointwise multiplication.
- Almost directly from the definitions, we get:
- **Theorem:** $C^{(h)}(L(a)) = L(h(a))$.



Composition of Correlation Matrices

- If $h(x)= f(g(x))$ then $C^{(h)}= C^{(f)}\ C^{(g)}$
- Proof
  - $(-1)^{u\bullet h(a)} = \sum_v C^{(f)}_{u,v} (-1)^{v\bullet g(a)}=\sum_v C^{(f)}_{u,v} (\sum_w C^{(g)}_{v,w}(-1)^{w\bullet a})$.

- If h is invertible, $(C^{(h)})^{-1}= (C^{(h)})^T$. Correlation matrices of invertible boolean transformations are thus orthogonal.
- Proof:
  - Let $g(y)= h^{-1}(y)$.
  - For a bijection, $C(u\bullet h^{-1}(a), w\bullet a)= C(u\bullet b, w\bullet h(b))= C(w\bullet h(b), u\bullet b)^T$, so, $C^{(g)}= (C^{(h)})^{-1}$

Invertible Boolean Transformations

- *Theorem:* A boolean transformation is invertible iff its correlation matrix is invertible.
  - The → direction follows from the inverse formula above.
  - The proof of ←: $(-1)^{u \bullet h(a)} = \sum_w C^{(h)}_{u,v} (-1)^{w \bullet a}$.
  - If $C^{(h)}_{u,v}$ is invertible,$(-1)^{w \bullet a} = \sum_u [(C^{(h)}_{u,v})^{-1}]_{w,u} (-1)^{u \bullet h(a)}$.
  - If exists $x \neq y$: $h(x) = h(y)$, substituting into the equation above, $(-1)^{w \bullet x}=(-1)^{w \bullet y}$ and that is just wrong.

Correlation matrices for standard functions

- Support: $V_f = \{w: F(w) \neq 0\}$. Result: $V_{f \oplus g} = V_f + V_g$.
- If $h(x) = x+k$, $C_{u,u} = (-1)^{u \bullet k}$
- If $h(x) = Mx$, $C_{u,w} = \delta(M^T u \oplus w)$.
- If $h(x) = (b_{(1)}, b_{(2)}, \ldots, b_{(n)})$, $b_{(i)} = h_{(i)}(a_{(i)})$ and $C^{(i)} = C^{h(i)}$ then $C_{u,w} = \prod_i C^{(i)}_{u(i),w(i)}$ (uses disjunct support).
- If $h(x) = g(x) + w \bullet x$, $H(u) = G(u \oplus w)$; if $V_f \cap V_g = \emptyset$, $w \in V_f$, $u \in V_g$, $H(u \oplus w) = F(w)G(u)$.

Correlation Matrix for Transposition

- $g(\mathbf{x}) = \Pi f(\mathbf{x})$ where $\Pi = (\mathbf{a}, \mathbf{b})$.
- $C^{(g)} = C^{(\Pi)} C^{(f)}$.
- $(C^{(\Pi)})_{uv} = 2^{-n} [\sum_{\mathbf{x} \neq \mathbf{a},\mathbf{b}} (-1)^{\mathbf{u \cdot x + v \cdot x}} + (-1)^{\mathbf{u \cdot b + v \cdot a}} + (-1)^{\mathbf{u \cdot a + v \cdot b}}]$
- $(C^{(\Pi)})_{uv} = 2^{-n} [\sum_{\mathbf{x}} (-1)^{\mathbf{u \cdot x + v \cdot x}} - (-1)^{\mathbf{u \cdot a + v \cdot a}} - (-1)^{\mathbf{u \cdot b + v \cdot b}} + (-1)^{\mathbf{u \cdot b + v \cdot a}} + (-1)^{\mathbf{u \cdot a + v \cdot b}}]$

$\mathbf{x_1,x_2,x_3}$	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
$f(x_1,x_2,x_3)$	000	001	101	011	100	010	110	111

- a= 010, b=101, u=010, v=011
- $(C^{(\Pi)})_{010,011} = 2^{-3}[-(-1)^0-(-1)^1+(-1)^1+(-1)^0] = -1+1-1+1=0$

- a= 010, b=101, u=100, v=001
- $(C^{(\Pi)})_{100,001} = 2^{-3} [-(-1)^0-(-1)^0+(-1)^1+(-1)^1] =(-1-1-1-1/)8= -0.50$

Example Correlation Matrix for

```
Calculate Correlation matrix of 3 bit Boolean transform: 0 1 5 3 4 2 6 7

000=u: 0 0 0 0 0 0 0 0
   000=v: 00000000 1.00000
   001=v: 01010101 0.00000
   010=v: 00110011 0.00000
   011=v: 01100110 0.00000
   100=v: 00001111 0.00000
   101=v: 01011010 0.00000
   110=v: 00111100 0.00000
   111=v: 01101001 0.00000
001=u: 0 1 1 1 0 0 0 1
   000=v: 00000000 0.00000
   001=v: 01010101 0.50000
   010=v: 00110011 0.50000
   011=v: 01100110 0.00000
   100=v: 00001111-0.50000
   101=v: 01011010 0.00000
   110=v: 00111100 0.00000
   111=v: 01101001 0.50000
```

Example Correlation Matrix for

```
Calculate Correlation matrix of 3 bit Boolean transform: 0 1 5 3 4 2 6 7

010=u: 0 0 0 1 0 1 1 1
   000=v: 00000000 0.00000
   001=v: 01010101 0.50000
   010=v: 00110011 0.50000
   011=v: 01100110 0.00000
   100=v: 00001111 0.50000
   101=v: 01011010 0.00000
   110=v: 00111100 0.00000
   111=v: 01101001-0.50000
011=u: 0 1 1 0 0 1 1 0
   000=v: 00000000 0.00000
   001=v: 01010101 0.00000
   010=v: 00110011 0.00000
   011=v: 01100110 1.00000
   100=v: 00001111 0.00000
   101=v: 01011010 0.00000
   110=v: 00111100 0.00000
   111=v: 01101001 0.00000
```

Example Correlation Matrix for

```
Calculate Correlation matrix of 3 bit Boolean transform: 0 1 5 3 4 2 6 7

100=u: 0 0 1 0 1 0 1 1
   000=v: 00000000 0.00000
   001=v: 01010101-0.50000
   010=v: 00110011 0.50000
   011=v: 01100110 0.00000
   100=v: 00001111 0.50000
   101=v: 01011010 0.00000
   110=v: 00111100 0.00000
   111=v: 01101001 0.50000
101=u: 0 1 0 1 1 0 1 0
   000=v: 00000000 0.00000
   001=v: 01010101 0.00000
   010=v: 00110011 0.00000
   011=v: 01100110 0.00000
   100=v: 00001111 0.00000
   101=v: 01011010 1.00000
   110=v: 00111100 0.00000
   111=v: 01101001 0.00000
```

Example Correlation Matrix for

```
Calculate Correlation matrix of 3 bit Boolean transform: 0 1 5 3 4 2 6 7

110=u: 0 0 1 1 1 1 0 0
   000=v: 00000000 0.00000
   001=v: 01010101 0.00000
   010=v: 00110011 0.00000
   011=v: 01100110 0.00000
   100=v: 00001111 0.00000
   101=v: 01011010 0.00000
   110=v: 00111100 1.00000
   111=v: 01101001 0.00000
111=u: 0 1 0 0 1 1 0 1
   000=v: 00000000 0.00000
   001=v: 01010101 0.50000
   010=v: 00110011-0.50000
   011=v: 01100110 0.00000
   100=v: 00001111 0.50000
   101=v: 01011010 0.00000
   110=v: 00111100 0.00000
   111=v: 01101001 0.50000
```

Example Correlation Matrix for

```
Calculate Correlation matrix of 3 bit Boolean transform: 0 1 5 3 4 2 6 7

Correlation Matrix (low order first):
 1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.500  0.500  0.000 -0.500  0.000  0.000  0.500
 0.000  0.500  0.500  0.000  0.500  0.000  0.000 -0.500
 0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000
 0.000 -0.500  0.500  0.000  0.500  0.000  0.000  0.500
 0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000
 0.000  0.500 -0.500  0.000  0.500  0.000  0.000  0.500
```

Multiplying Correlations Matrices

- Theorem: $C^{(h)}_{u\oplus v,x} = \sum_w C^{(h)}_{u,w\oplus x}\, C^{(h)}_{v,w}$.
- Proof:
  - $\mathcal{W}((u\oplus v)\bullet h(a)) = \mathcal{W}(u\bullet h(a))\otimes(v\bullet h(a))$;
  - Note that first transform on right is $C^{(h)}_{u,w}$ and second is $C^{(h)}_{v,w}$. One consequence is: $C_{u\oplus v,0} = \sum_w C_{u,w}\, C_{v,w}$

- If u and w are parities then and F^u denotes the normalized Walsh transform of **u•f**, while G^w denotes the normalized Walsh transform of **w•g**(**x**) then $(C(\mathbf{f},\mathbf{g}))_{u,w} = \sum_v F^u(v)\, G^w(v)$.

Correlation matrix for invertible transformations

- *Theorem:* A Boolean transformation is invertible iff every output parity is a balanced binary boolean function of the input bits.
- Proof
  - →: If h is invertible, $C\,C^T = I$, $C_{00}=1$ and the norm of every row and column is 1. $C(u \bullet h(a),0) = \delta(u)$; all rows except row 0 are correlated to 0 hence the function is balanced for $u \neq 0$. $\sum_v F^u(v)\, G^w(v)$.
  - ←: The condition on output parities being balenced is $C_{u,0}=0$, $u \neq 0$. i.e.- C is orthogonal. $C\,C^T = I$ $\sum_w C_{u,w}\, C_{v,w} = \delta(u \oplus v)$ (*) also $\sum_w C_{u,w}\, C_{v,w} = C_{u \oplus v,0}$ but $C_{u,0}=0$, $u \neq 0$ and $C_{00}=1$ so * holds for all u,v hence C is orthogonal. Let **f** and **g** be two surjective boolean transformations on n variables and define C(**f**, **g**) in the obvious way.

Possible Spectrums

- *Theorem:* The correlation coefficients and spectrum values for a boolean function over GF(2) are integer multiples of 2^{1-n}.

  - Proof: Let h[r]= $h^{(r)}$. The values are of the form $k+(2^n-k)(-1)=2k-2^n$ which is even. Given $f:GF(2)^n \rightarrow GF(2)^m$, let the restriction to n-1 bits be specified by $v^T \cdot a = \square$ modelled by $a'=h^{(r)}(a)$, $a_i'=a\_i$ if $i \neq s$ and $a_s'= \epsilon \oplus v \bullet a \bullet a_s$. $C_{w,w}^{h[r]}=1$, $C_{v \oplus w,w}^{h[r])}=(-1)^{\square}$, if $w_s=0$, $C'_{u,w}= C_{u,w \oplus v}+(-1)^{\square} C_{v,w}$, $w_s= 0, 0$ if $w_s= 1$. $\square \square_w (F(w) F(w \oplus v))^2 =2$. Colliding pairs are rare (probability is 2^{-nk})

Constructing Boolean Transformations

- Each possible Boolean transformation on n bits is a permutation on the 2^n, n-bit values and so listing them in order, the columns are the possible **f** vectors representing the component functions.

- If we label these as points in $GF(2)^N$ and draw an edge for allowable co-components with the edges labelled by the correlation between these vectors, any allowable n boolean functions form a complete graph with the label 0 on each edge.

More properties of correlation entries

- Let $N=2^n$. *Theorem:* The elements of a correlation matrix corresponds to an invertible transform of n-bit vectors are integer multiples of 2^N.
  - The proof uses the restriction map and the fact that $\square$ $(F(w)+F(w+v))^2 = 2$.

- All correlation matrices are doubly stochastic.
- Correlation matrices for involutions are symmetric.
- $W(u \bullet h)= \otimes_{u[i]=1} H_i$.

Relationships among invertible transformation components

- Suppose F: $GF(2)^n \rightarrow GF(2)^n$ is a bijection and $f_i = \phi_i(F)$ then $C(f_i,0) = C(f_j,1) = 0 = C(f_i, f_j)$. $wt(f_i) = 2^{n-1}$, $wt(f_i f_j) = 2^{n-2}$, etc.
- $C(f_i f_j, f_k) = ½$, $C(f_i f_j f_k, f_l) = C(f_i f_j, f_k f_l) = C(f_i f_j f_k, f_l)$.

- *Theorem 1:* $C(f_i,1) = C(f_i,0) = 0$, $C(f_i, f_j) = 0$, $i \neq j$, $wt(f_i) = 2^{n-1}$, for all i, $wt(f_i f_j) = 2^{n-2}$, $i \neq j$ and in general, $wt(f_{i1} f_{i2} \ldots f_{ik}) = 2^{n-k}$. Further, $C(f_i f_j, f_k) = 1/2$, $C(f_i, f_j, f_k f_l) = C(f_i f_j f_k, f_{il})$ and in general $C(f_{i1} f_{i2} \ldots f_{ik}, f_l) = 2^{n-k-1}$.
- *Theorem 2:* Let f be a boolean function. The N functions $f_{i1}, f_{i2} \ldots f_{ik}$ form a basis for the space of boolean functions; that is, for any boolean function g, exists $a^{(g)}_{i1, i2, \ldots, ik}$ such that $g(x) = \sum_{1 \leq i1 < i2 < \ldots < ik = n} a^{(g)}_{i1, i2, \ldots, ik} f_{i1} f_{i2} \ldots f_{ik}$. In particular, there are such coefficients such that $x_i = \sum_{1 \leq i1 < i2 < \ldots < ik = n} a^{(xi)}_{i1, i2, \ldots, ik} f_{i1} f_{i2} \ldots f_{ik}$.

- Define $Appx_i(f) = \{g: dist(f,g) \leq i\}$, then $|Appx_i(f)| = \sum_{j=0}^{I} {}_N C_i$.

Classifying boolean functions

- Let f,g: $GF(2)^n \rightarrow GF(2)$. f and g are said to be *affinely equivalent* if $f(M_1\mathbf{x})+M_2\mathbf{x}=g(\mathbf{x})$ for invertible linear transformations M_1 and M_2.
- The spectra of affinely equivalent functions have the same absolute values.
- Affine equivalence induces an equivalence relation among the set of boolean functions.
- RM(1,5) has 48 inequivalent affine classes for example.

Bent Functions

- Bent functions are furthest from linear.

- All Hadamard transform values of bent functions are equal to $\pm 2^{m/2}$ and hence the distance to any affine function is $2^m \pm 2^{m/2}-1$.
- If $f(x_1, x_2, ..., x_m)$ is bent and $m \geq 6$ then f is indecomposable.
- $f(u_1, u_2, ..., u_m, v_1, v_2, ..., v_m) = g(v_1, v_2, ..., v_m) + \sum_i u_i v_i$ are bent.
- If $f(u_1, u_2, ..., u_m, v_1, v_2, ..., v_m) = \sum_i u_i v_i$, then $f+u_1u_2u_3$, $f+u_1 u_2u_3u_4$, ...,$f+u_1 u_2u_3... u_m$ are all inequivalent bent functions

How many Boolean Matrices are invertible

- Let r_n be the ratio of the number of invertible matrices to the number of matrices. r_n approaches .288 and n $\rightarrow\infty$.
  - Proof:
  - The number of boolean matrices is 2^N, $N= n^2$.
  - The number of invertible matrices is $t_n= (2^n-1)(2^n-2)\ldots(2^n-2^{n-1})$.
  - $t_n= 2^M(2^n-1)(2^{n-1}-1)\ldots(2-1)$ where $M=(n(n-1))/2$.
  - Define $s_n= (2^n-1)(2^{n-1}-1)\ldots(2-1)$.
  - Note that $t_{n+1}= 2^{M'} s_{n+1}= 2^{M'} s_n (2^{n+1}-1)$ where $M'=(n(n+1))/2$. As a result, $t_{n+1}= 2^{M'} 2^{-M} (2^M s_n)(2^{n+1}-1)= 2^{M'-M} t_n (2^{n+1}-1)= 2^n(2^{n+1}-1) t_n$.
  - Combining these we get, $r_{n+1}= t_{n+1}/2^{N'}= 2^n(2^{n+1}-1)(t_n/2^N)2^{N'-N}$, where $N'= (n+1)^2$.
  - So $r_{n+1}= r_n(2^{n-(2n+1)} (2^{n+1}-1)= r_n(1-2^{-(n+1)})$.
  - Using this recurrence: $r_n= \prod_{i=1}^{n} (1-2^{-n})$.
  - The product approached 0.288---.

Orthogonal Transformations

- Since the Walsh transform determines the best linear approximator of a function, so the correlation matrix gives the best linear approximation among any linear combination of the components of a boolean transformation.

- Here is a motivating example in R^3:

- $R = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix}$

$$R^{-1}TR = \begin{pmatrix} \cos^2(\theta)+\cos(\theta)\sin^2(\theta) & \cos(\theta)\sin(\theta)-\cos(\theta)\cos(\theta)\sin(\theta) & -\sin(\theta)\sin(\theta) \\ -\cos(\theta)\sin(\theta)+\cos(\theta)\cos(\theta) & \sin(\theta),\sin^2(\theta)+\cos(\theta)\cos^2(\theta) & \sin(\theta)\sin(\theta) \\ \sin(\theta)\sin(\theta) & -\cos(\theta)\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Feistel transformations

- A typical round of DES consists of two involutions: σ and τ_k. $\tau_k(L,R) = (L \oplus f(R,k), R)$, $f(\mathbf{x},\mathbf{k}) = P\ S_1\ S_2\ \ldots\ S_8\ (E(\mathbf{x})+\mathbf{k})$. $\sigma(L,R) = (R,L)$.

- First line of τ_k is
  - $y_9 = x_9 \oplus S_1^1(x_{64}+k_1, x_{33}+k_2, x_{34}+k_3, x_{35}+k_4, x_{36}+k_5, x_{37}+k_6)$
  - $y_{17} = x_{17} \oplus S_1^2(x_{64}+k_1, x_{33}+k_2, x_{34}+k_3, x_{35}+k_4, x_{36}+k_5, x_{37}+k_6)$
  - $y_{23} = x_{23} \oplus S_1^2(x_{64}+k_1, x_{33}+k_2, x_{34}+k_3, x_{35}+k_4, x_{36}+k_5, x_{37}+k_6)$
  - $y_{31} = x_{31} \oplus S_1^2(x_{64}+k_1, x_{33}+k_2, x_{34}+k_3, x_{35}+k_4, x_{36}+k_5, x_{37}+k_6)$

Calculating correlation for DES

- If a transformation is a composition of a sequence of transformations, the correlation matrix of DES is a product of the per round function correlation matrices.

- To calculate the round correlation for DES, decompose it into three involutions.
  - The first, adds output from odd numbered S-boxes but is otherwise the identity. The second, adds output from even numbered S-boxes but is otherwise the identity.
  - The third transposes L and R.
  - The first and second involutions don't overlap on input variables to the S-boxes so the Walsh transforms of components of the S-boxes are all that is needed.
  - In both the first and second transformations, each position affected by an S-box is multiplied by $(-1)^{w \bullet k}$ (i.e. - ±1) for the relevant round keys.

Calculating correlation for DES

- $\pi(L,R) = (R,L)$
- Let $T_i(kr, x) = S_i[(E(x)+kr)_{6(i-1)+1\ldots6i}]$.
- $\sigma_{kr}^{1}(L,R) = L \oplus (T_1(kr, R), 0, T_3(kr, R), \ldots, T_7(kr, R), 0)$
- $\sigma_{kr}^{2}(L,R) = L \oplus (0, T_2(kr, R), 0, T_4(kr, R), \ldots, T_8(kr, R))$
- $\phi_i(L,R) = \pi(\sigma_{kr}^{2}(\sigma_{kr}^{1}(L,R)))$ is equation for round I of DES.
- Calculate the correlation matrix for each of the three transformations and multiply them together.

Correlation Matrix for …

- Let $f(x_1,x_2,x_3,x_4)= (x_1+f_1(x_3,x_4), x_2+f_2(x_3,x_4),x_3,x_4)$.
- $h(x_3,x_4)= f_1(x_3,x_4)+f_2(x_3,x_4)$.

$C^{(f)}=$

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	$F_2(0)$	$F_2(1)$	$F_2(2)$	$F_2(3)$	0	0	0	0	0	0	0	0
0	0	0	0	$F_2(1)$	$F_2(0)$	$F_2(3)$	$F_2(2)$	0	0	0	0	0	0	0	0
0	0	0	0	$F_2(2)$	$F_2(3)$	$F_2(0)$	$F_2(1)$	0	0	0	0	0	0	0	0
0	0	0	0	$F_2(3)$	$F_2(2)$	$F_2(1)$	$F_2(0)$	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	$F_1(0)$	$F_1(1)$	$F_1(2)$	$F_1(3)$	0	0	0	0
0	0	0	0	0	0	0	0	$F_1(1)$	$F_1(0)$	$F_1(3)$	$F_1(2)$	0	0	0	0
0	0	0	0	0	0	0	0	$F_1(2)$	$F_1(3)$	$F_1(0)$	$F_1(1)$	0	0	0	0
0	0	0	0	0	0	0	0	$F_1(3)$	$F_1(2)$	$F_1(1)$	$F_1(0)$	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	H(0)	H(1)	H(2)	H(3)
0	0	0	0	0	0	0	0	0	0	0	0	H(1)	H(0)	H(3)	H(2)
0	0	0	0	0	0	0	0	0	0	0	0	H(2)	H(3)	H(0)	H(1)
0	0	0	0	0	0	0	0	0	0	0	0	H(3)	H(2)	H(1)	H(0)

Correlation Matrix for Swap

- Define $\Pi(x_1, x_2, x_3, x_4) = (x_3, x_4, x_1, x_2)$
- $C^{(\Pi)} =$

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Correlation Matrix f

- $C^{(f)}$=

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	F2(0)	F2(1)	F2(2)	F2(3)	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	F1(0)	F1(1)	F1(2)	F1(3)	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	H(0)	H(1)	H(2)	H(3)
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	F2(1)	F2(0)	F2(3)	F2(2)	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	F1(1)	F1(0)	F1(3)	F1(2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	H(1)	H(0)	H(3)	H(2)
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	F2(2)	F2(3)	F2(0)	F2(1)	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	F1(2)	F1(3)	F1(0)	F1(2)	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	H(2)	H(3)	H(0)	H(2)
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	F2(3)	F2(2)	F2(1)	F2(0)	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	F1(3)	F1(2)	F1(1)	F1(0)	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	H(3)	H(2)	H(1)	H(0)

Linear trails

- A *linear trail* is $U= (u^{(0)}, u^{(1)}, ..., u^{(r)})$ associated with a composite function $\pi= \pi^{(0)} \pi^{(1)} ... \pi^{(r)}$ with correlation contribution at each step of $C((u^{(i)})\bullet\pi^{(i)}(a), u^{(i-1)}\bullet a)$ and overall correlation of $C^p(U)= \Pi_i [C^{\pi(0)}]_{u(i), u(i-1)}$.

- *Theorem:* $C(u\bullet\pi(a), w\bullet a)= \Sigma_{U, u(0)=u, u(r)=w} C_p(U)$.

- *Truncating Function:* Let $a'=h^{(r)}(a)$, $h[r]=h^{(r)}$ and $h[r]:GF(2)^{n-1} \rightarrow GF(2)^n$ be defined by $a_i'=a_i$ for $i\neq s$ and $a_s'= \epsilon\oplus v\bullet a\oplus a_s$ where $v^T a=$ [illegible] defined the restriction. Then
  - $C^{h[r]}_{w,w}=1$
  - $C^{h[r]}_{v\oplus w,w}= (-1)^{\pi}$, for all w: $w_s=0$; note there are the non-zero entries both of amplitude 1.
  - If $C'= C\, C^{h(r)}$, $C'_{u,w}= C_{u,w}\oplus(-1)^{\pi} C_{u,v\oplus w}$ if $w_s=1$ and 0 if $w_s=0$.

Long Range correlation

- Put u[i]= $u^{(i)}$, k[i]=k(i), D[U]= $d_U \oplus \oplus_i (u^{(i)})^T k^{(i)}$, s[i]= s_i, D[U,K]= $d_U \oplus U^T K$.

- For key alternating ciphers, $C_p(U) = \Pi_i (-1)^{D[U]} |C_p(U)|$.

- Put $s_i = U^T K \oplus d_U$, $C(v \bullet \Pi(a), w \bullet a) = \Pi_{U,\, u(0)=u,\, u(r)=w} (-1)^{D[U,K]} |C_p(U)|$.

- $C_p(U) = (-1)^{s[i]} C_i$, averaging over the round keys we get $E(C_t^2) = 2^{-nk} \Pi_k (\Pi_i (-1)^{s[j]} C_i)^2$.

- After reduction, average correlation potential is $E(C_t^2) = \Pi_i C_i^2$, note that $C_i C_j = 2^{nK} \Pi(i \oplus j)$

Key Schedule and Correlation

- Let U[j]= U_j, d[U,j]= $d_{U[j]}$, h[r]= $h^{(r)}$. C[h,r]= $C^{h[r])}$.
  $\Box$ = (d[U,i]⊕d[U,j])^T $M_{C[h,r]}$k⊕d[U,i]⊕d[U,j]. $\Box$ = (d[U,i]⊕d[U,j])^T $f_{\Box}$(k)⊕d[U,i]⊕d[U,j].

- For key schedule K=$M_{\Box}$**k**,
  - $E(C_t^2)= 2^{-nK} \sum_i \sum_j (\sum_k (-1)^{\Box}) C_i C_i$.
- The inner sum simplifies to $(-1)^{d[U,i]\oplus d[U,j]} 2^{nK} \Box(M_{\Box}^T(U_i \oplus U_i))$.
- f key schedule is not linear K=$f_{\Box}$(k), the coefficient of the mixed term is $(-1)^{\Box}$.
- The probability that a multi-round expression holds is $1/2(1+C_p(U))$ for the associated trail

Take home on linear propagation

- Correlation matrix completely determines linear propagation.
- Individual round as composition of key xor, linear and bricklayer functions are easy to compute.
- Linear trails provide link between individual approximations and full cipher.
- Key schedule only effects sign of contribution.
- Keys select constructive or destructive interference.
- Most reasonable key schedules provide destructive interference.
- The probability that a multi-round expression holds is $1/2(1+C_p(U))$ for the associated trail.

Differentials

- A similar theory applies to differentials.
- *Definition:* The difference propagation probability, denoted by $R_p(a' \rightarrow_h b')$, is defined by
  $$\text{Prob}^h(a',b') = 2^{-n} \sum_a \delta(b' \oplus h(a \oplus a') \oplus h(a)).$$
- We have $0 \leq R_p(a' \rightarrow_h b') \leq 1$. $w_r(a' \rightarrow_h b') = -\lg(R_p(a' \rightarrow_h b'))$ (restriction weight reflect loss of entropy).
- $w_c(U) = -\lg(|C_p(U)|)$ (correlation weight).
- For bricklayer function, $\text{Prob}^h(a', b') = \prod_i \text{Prob}^{h(i)}(a'_{(i)}, b'_{(i)})$ and $w_r(a', b') = \sum_i w_r(a'_{(i)}, b'_{(i)})$.

Differential trails

- *Theorem:* $\text{Prob}^f(a', 0) = 1/2\,(1 + \sum_w (-1)^{w \bullet a'} F(w)^2)$. The differential probability and correlation potential table of a boolean function satisfy $\text{Prob}(a', b') = 2^{-m} \sum_{u,\,w} (-1)^{w \bullet a' \oplus u \bullet b'} C_{u,w}^2$

- A *differential trail* is $Q = (q^{(0)}, q^{(1)}, \ldots, q^{(r)})$ with steps $(q^{(i-1)}, q^{(i)})$ having weight $w_r^{(i)}(q^{(i-1)}, q^{(i)})$ have trail weight $w_r(Q) = \sum_i w_r^{(i)}(q^{(i-1)}, q^{(i)})$.
- $\text{Prob}(a', b') = \sum_{q(i-1)=a',\, q(r)=b'} \text{Prob}(Q)$.
- For a differential trail, Q, with weight <(n-1),

  $\text{Prob}(Q) \sim 2^{-wr(Q)}$.
- For a differential trail, Q, with weight $w_r(Q)>(n-1)$, for expected proportion $2^{n-1-wr(Q)}$ of keys, there will be a right pair.

Take home on differential propagation

- Correlation matrix completely determines differential propagation characteristics.
- Individual round as composition of key xor, linear and bricklayer functions are easy to compute.
- Differential trails provide link between individual approximations and full cipher.
- Weights for differential trails are good approximation for differential characteristics.

Rijndael Design Principles - motivation

- The theory of linear and differential trails informed the design of Rijndael.

- To eliminate low weight trails, there are two strategies:
  1. Choose S-boxes with difference propagations that have high restriction weight and input-output correlations with high correlation weights; or,
  2. Design round transformations so that only trails with many S-boxes occur.
- Rijndael picks 2.
- Wide trails strategy implements this.

Rijndael Design Principles - continued

- *Linear cryptanalysis* requires correlation > $2^{-nb/2}$ over most rounds. This can't happen if we choose the number of rounds so that there are no such linear trails with correlation contribution >nk^{-1} $2^{-nb/2}$. Each output parity is correlated to an input parity since $\sum_w F(w)^2=1$ but if it occurs by constructive interference over many trails that share input/output selection then any such must be the result of at least nk linear trails which are unlikely to be key dependent.
- *Differential cryptanalysis* requires input to output difference propagation with probability >2^{1-nb}. If there are no differential trails with low weight, difference propagation results from multiple trails which again will not likely be key dependent.

Rijndael Design Principles

- Choose number of rounds so that there is no correlation over all but a few rounds with amplitude significantly larger than $2^{nb/2}$ by insuring there are no linear trails with correlation contribution above $nk^{-1}\ 2^{nb/2}$and no differential trails with weight below nb.

- Rijndael also insures that the diffusion layer provides that no multiple round trails have few active S-boxes. This guarantees no iteratively constructed correlation exists over several rounds.

Amplitudes

- Examine round transformations □= □□, where □ is the mixing function and □ is a bricklayer function that acts on bundles of nt bits. Block size is nb=m nt.
- The correlation over □ is the product of correlations over different S-box positions for given input and output patterns.
- Define weight of correlation as -lg(Amplitude).
- If output selection pattern is ≠0, the S-box is active. Looking for maximum amplitude of correlations and maximum difference propagation probability.
- The weight of a trail is the sum of the weights of the selection patterns or the sum of the active S-box positions it is greater than the number of active S-boxes times the minimum correlation weight per S-box.
- Wide trail: Design round transformations so there are no trails with low bundle weight.

Branching and wide trails

- Define $w_b(a)$ as the bundle weight of a. Let $C(\alpha, \beta, \gamma, x)$= $\alpha, \beta, C(\alpha \bullet x, \beta \bullet \gamma(x)) \neq 0$.
- $\mathcal{B}_d(\theta) = \min_{a,\, b \neq a} (w_b(a \oplus b) + w_b(\theta(a) \oplus \phi(b)))$.
- $\mathcal{B}_l(\alpha, \beta) = \min_{C(\alpha, \beta, \gamma, x)} (w_b(\alpha) + w_b(\beta))$.

- Theorem: In an alternating key block cipher with $\geq$ 2 round functions, the number of active bundles in a two round trail is $\geq$ the bundle branch number of θ. If $\theta = \theta_4 \circ \theta_3 \circ \theta_2 \circ \theta_1$ is a four round function, $\mathcal{B}(\theta) \geq \mathcal{B}(\theta_1) \times \mathcal{B}^c(\theta_3)$ where $\mathcal{B}$ can be either the linear or differential branch number.

- The linear and differential branch numbers for an AES round is 5.

Rijndael local safety results

- No 4 round differential occurs with probability greater than 2^{-150}.
- No 8 round differential occurs with probability greater than 2^{-300}.
- No 4 round I/O correlation occurs with probability greater than 2^{-75}.
- No 8 round I/O correlation occurs with probability greater than 2^{-150}.

Rijndael diffusion safety results

- 4 round versions have more than 25 active S-boxes.
- The weight of a two round differential trail with Q active columns at the input of the second round is $\geq 5Q$.
- In a two round trail, the sum of the active columns at the input and output is ≥ 5.
- Net effect is that there are not enough pairs in the I/O of Rijndael to permit a linear or differential attack in time better than exhaustive search.

- Best 14 round DES correlation is $\frac{1}{2} \pm 1.19 \times 2^{-21}$.

End

Some example functions

- $a \vee b = a \oplus b \oplus ab$ as a boolean function.
- Let $\mathbf{x} = (x_4, x_3, x_2, x_1)$ with x_1 the least significant bit.
- $\mathbf{F}(\mathbf{x}) = (F_4(\mathbf{x}), F_3(\mathbf{x}), F_2(\mathbf{x}), F_1(\mathbf{x}))$.
- If $\Gamma = (0000, 0001)$ then $\mathbf{F}_i^{\Gamma}(\mathbf{x}) = x_i$, $i>1$ and
- $\mathbf{F}_i^{\Gamma}(\mathbf{x}) = (\neg(x_2 \vee x_3 \vee x_4)\ (x_1 \oplus 1) \oplus (x_2 \vee x_3 \vee x_4)\ x_1 = 1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_2x_3 \oplus x_2x_4 \oplus x_3x_4 \oplus x_2x_3x_4$
- If $\Gamma = (0000, 0001, \ldots, 1111)$, then
  - $\mathbf{F_1}^{\Gamma}(\mathbf{x}) = x_1 \oplus 1$,
  - $\mathbf{F_2}^{\Gamma}(\mathbf{x}) = x_1(x_2 \oplus 1) \oplus (\neg x_1)x_2 = x_1 \oplus x_2$,
  - $\mathbf{F_3}^{\Gamma}(\mathbf{x}) = (x_1x_2)\ (x_3 \oplus 1) \oplus (\neg(x_1x_2)))x_3 = x_1x_2 \oplus x_3$,
  - $\mathbf{F_4}^{\Gamma}(\mathbf{x}) = (x_1x_2x_3)\ (x_4 \oplus 1)$
  - $\mathbf{F_4}^{\Gamma}(\mathbf{x}) = (x_1x_2x_3)\ (x_4 \oplus 1) \oplus (\neg(x_1x_2x_3))x_4 = x_1x_2x_3 \oplus x_4$.

Ideas to study

- Suppose the Boolean Transformation: Is there an easy to compute function, T_K, obviously non-linear, so that $T_K E_K T_K^{-1}$ has good linear approximations?
- How do you find such T_K?
- Finding the best approximation reduces to finding an orthogonal transformation that maximizes the largest entry. Suppose T is such a matrix; if T has all bad affine approximations
- is it possible that there is another orthogonal transformation, R with
- $T^R = R^{-1} T R$ such that $\max_{ij}(|(T^R)_{ij}|) > \max_{ij}(|(T)_{ij}|)$?
- If $\pi_1, \pi_2, \ldots, \pi_n$ is a series of such transformations (like the iterated components of a block cipher), note that $R^{-1}E_K(x) R = R^{-1}\pi_1 R\ R^{-1}\pi_2 R \ldots R^{-1}\pi_n R$ thus raising the possibility of better per round approximations on a related cipher.

Correlations and AES

- $Tr(C^{(AES)})$ is the number of fixed points of AES.

- Since Tr(AB)=Tr(BA),
- $Tr(C^{AES)})= Tr(C^{k14}) C^{(k13)}) ... C^{(k1)} C^{(RS)} (C^{(MRS)})^{13})$.

- $NL(f) \leq 2^{n-1} - 2^{n/2 -1}$,
- $NL(f) \leq 2^{n-1}+\sqrt{(2^n + \max_{e\neq 0} (F(D_e(f)))}$, where $D_e f = f(x) \oplus f(x \oplus e)$.
- What does eigenvalue of correlation matrix mean?
- If λ is an eigenvalue, $\lambda^2=1$.
- When is a correlation matrix blocky?

Correlations and AES

- $Tr(C^{(AES)})$ is the number of fixed points of AES.

- Tr(AB)=Tr(BA):
- $Tr(C^{AES)}) = Tr(C^{k14}) C^{(k13)}) ... C^{(k1)} C^{(RS)} (C^{(MRS)})^{13})$.

- $NL(f) \leq 2^{n-1} - 2^{n/2 -1}$,
- $NL(f) \leq 2^{n-1} + \sqrt{(2^n + \max_{e \neq 0} (F(D_e(f)))}$, where $D_e f = f(x) \oplus f(x \oplus e)$.

The Trace

- Let $e(i)= 2^i$.
- For F_q, $q=2^n$, $Tr_{Fq/F2}(x)=Tr(x)= \sum_{i=0}^{n-1} x^{e(i)}$.

- Theorem: $Tr(x) \neq 0$ for some x.
  - $Tr(x+y)= Tr(x)+Tr(y)$.
  - $Tr(x^2)= Tr(x)$.
  - $Tr(x)$ in F_2.
  - $Tr(\alpha \cdot x)$ is linear in x.
  - $Tr(\alpha_1 \cdot x)= Tr(\alpha_2 \cdot x) \rightarrow \alpha_1 = \alpha_2$.
  - $Tr(\alpha \cdot x)$ are exactly the linear functions.

Distance between functions

- $NL(f) \leq 2^{n-1}-2^{n/2-1}$, $NL(f) \leq 2^{n-1} + \sqrt{(2^n + \max_{e\neq 0} (F(D_e(f)))}$, where $D_e f = f(x)\oplus f(x\oplus e)$.

- Theorem (Rothaus): Let $n\geq 4$ of even algebraic degree then any bent function on $GF(2)^n$ has degree $\leq n/2$. An n-Boolean function, f, is m-resilient iff f is balanced and $F(u)=0$, for all u: $wt(u) \leq m$.
-
- Maiorana-MacFarland class $\mathcal{M} = \{f: f(x,y)=x \pi(y)\oplus g(y)\}$ where π is a permutation on $GF(2)^{n/2}$ and g is affine.

- $|\mathcal{M}|= (2^{n/2})!\ 2^{n/2}$
- For Bent Quadratics: $\oplus_{1\leq i,j \leq n} a_{ij} x_i x_j \oplus h(x)$, h, affine.

Correlation Immunity

- In this paragraph, F denotes the unnormalized Walsh transform of f.
- A function $z=f(x_1, x_2, ..., x_n)$ on n variables $x_1, x_2, ..., x_n$ is m-th order correlation immune if for every subset of these variables or size m, $I(z; x_{i1}, ..., x_{im})=0$. Equivalently, f is correlation immune of order m: $F(\omega)=0$ $\forall \omega: 1 \le wt(\omega) \le m$.
- If f has correlation immunity m and non-linear order k, $m+k \le n$, let $N_{ab}(\omega)= | \{ x: z=f(x)=a, \omega \bullet x = b \} |$ then $F(\omega)= N_{10}(\omega) - N_{11}(\omega)$.
- Denote $p_a = P(z=a)$ then $P(\omega \bullet x = b \mid z=a)=P(\omega \cdot x=b, z=a)/P(z=a)= p_a^{-1} 2^{-n} N_{ab}(\omega)$.
- We obtain the following:
  - $P(\omega \cdot x=0 \mid z=1)= \frac{1}{2} + p_1^{-1} 2^{-n-1} F(\omega)$,
  - $P(\omega \cdot x=1 \mid z=1)= \frac{1}{2} - p_1^{-1} 2^{-n-1} 2^{-n-1} F(\omega)$,
  - $P(\omega \cdot x=0 \mid z=0)= \frac{1}{2} + p_0^{-1} 2^{-n-1}\} 2^{-n-1} F(\omega)$,
  - $P(\omega \cdot x=1 \mid z=0)= \frac{1}{2} - p_0^{-1} 2^{-n-1}\} 2^{-n-1} F(\omega)$.
- Let $h(t)= - t \lg(t) - (1-t) \lg(1-t)$.

Correlation Immunity based attack

- s

Algebraic Immunity

- Low degree approximations exists $g \neq 0$: $fg = 0$ and fg has low degree $\deg(fg) \geq \deg(f)$. $|S_d| = \sum_{i=0}^{d} {}_nC_i$.

- Let f be a boolean function of n variables. The annihilator ideal of f, $AN(f) = \{g: g(x) f(x)=0\}$, for all x in $GF(2)^n$, $AN_d(f) = \{g \in AN(f): \deg(g(x)) \leq d\}$.

- The algebraic immunity, AI(f) is the smallest degree non-zero polynomial in $AN(f) \cup AN(1+f)$. $AI(f) \leq [n/2]$.

Shift registers and immunity

- Suppose $\mathcal{L}$ is an n-bit NLFSR based filter generator with filter function f and that L takes the current n-bit state to the next n-bit state. Suppose the initial state is $\mathbf{x_0}$, the generated keystream is $s_t = f(L^t(\mathbf{x_0}))$. $s_t=1$ if $\exists g \in AN_d(f)$: $g(L^t(\mathbf{x_0}))=0$, $s_t=0$ if $\exists h \in AN_d(1+f)$: $h(L^t(\mathbf{x_0}))=0$.

- Collect all functions of degree $\leq d$ for N known keystream bits; then,
  1. $g(L^t(x_1, x_2, ..., x_n))$: $\exists g \in AN_d(f)$, forall $0 \leq t < N$: $s_t=1$; and,
  2. $h(L^t(x_1, x_2, ..., x_n))$: $\exists g \in AN_d(1+f)$, $\exists 0 \leq t< N$: $s_t=0$.

- Using linearization to solve these equations requires identifying the subset of monomials forming a linear system of up to $\sum_{i=1}^{d} {}_nC_i$ variables.
- Gaussian reduction on this system takes time $O((\sum_{i=1}^{d} {}_nC_i)^{\omega})\sim n^{\omega d}$ where $\omega \sim 2.37$ and the number of monomials is

  $\sim 2n^d/(d!(\dim(AN_d(f))+ \dim(AN_d(1+f)))$.

Sensitivity

- For this section, f:GF(2)^m →GF(2). The sensitivity of v is defined by S(v) = |{ v': f(v)≠f(v'), dist(v,v‘)=1}|. The average sensitivity S(f)= 2^{-m} $\sum_v$ S(v). The “influence'' of x_i is defined by I(x_i)= Prob(f(x_1,...,x_{i-1}, y, x_{i+1},..., x_m), the probability that the function is determined no matter what y is.
- *Theorem:* Let f be a boolean function of n variables with average sensitivity aS(f)=k. Let ε>0 and M= k/ε then
  - $\exists$ h depending on exp((2+√((2 log(4M))/ M) M) variables such that Prob(f≠h)≤ε; and,
  - $\exists$ g of degree at most exp((2+√((2 log(4M))/ M))M) such that Prob(f≠g)≤ε/2.