On differential patterns for attacks on SHA-1

Krystian Matusiewicz and Josef Pieprzyk

`kmatus@ics.mq.edu.au, josef@ics.mq.edu.au`

Centre For Advanced Computing, Algorithms and Cryptography,

Department of Computing,

Macquarie University

Talk overview

- Cryptographic hash functions : basic notions
- Descriptions of SHA-0 and SHA-1
- Differential attack of Chabaud and Joux on SHA-0
- Finding patterns for attacks on variants of SHA-1
- Experimental results
- Some bounds on weights of short patterns

Cryptographic hash functions

Hash function - a function that maps binary strings of arbitrary length to strings of fixed length,

$$h : \{0,1\}^* \to \{0,1\}^n.$$

Cryptographic hash function - hash function with additional properties:

- fast to compute
- preimage resistant
- second preimage resistant
- collision resistant

Properties of cryptographic hash functions

Preimage resistant : Given an output Y of the hash function it is difficult to find any *preimage* - an input X such that $h(X) = Y$.

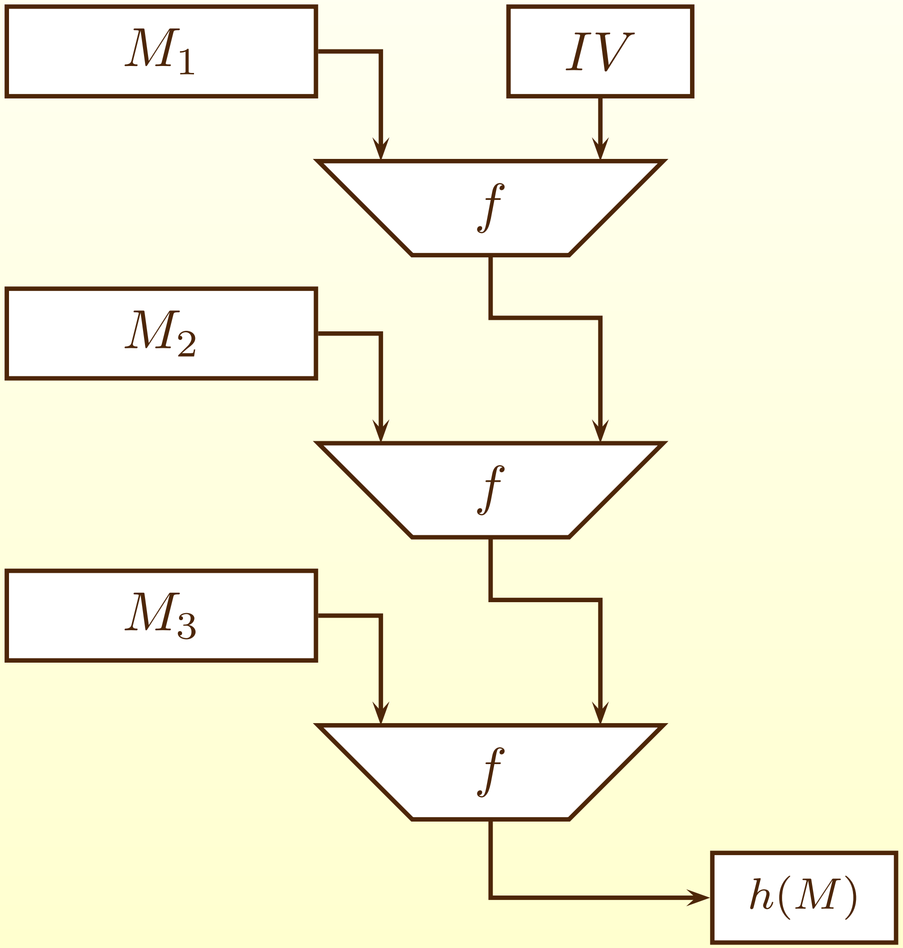
Second preimage resistant : Given a fixed input X to the hash function and corresponding output $h(X)$ it is difficult to find a *second preimage* - another input X', $X' \neq X$ such that $h(X) = h(X')$.

Collision resistant : It is hard to find any pair of distinct messages (X, X'), $X \neq X'$ such that $h(X) = h(X')$.

Attack on a hash function: finding a preimage or a collision.

Iterative hash functions from compression functions

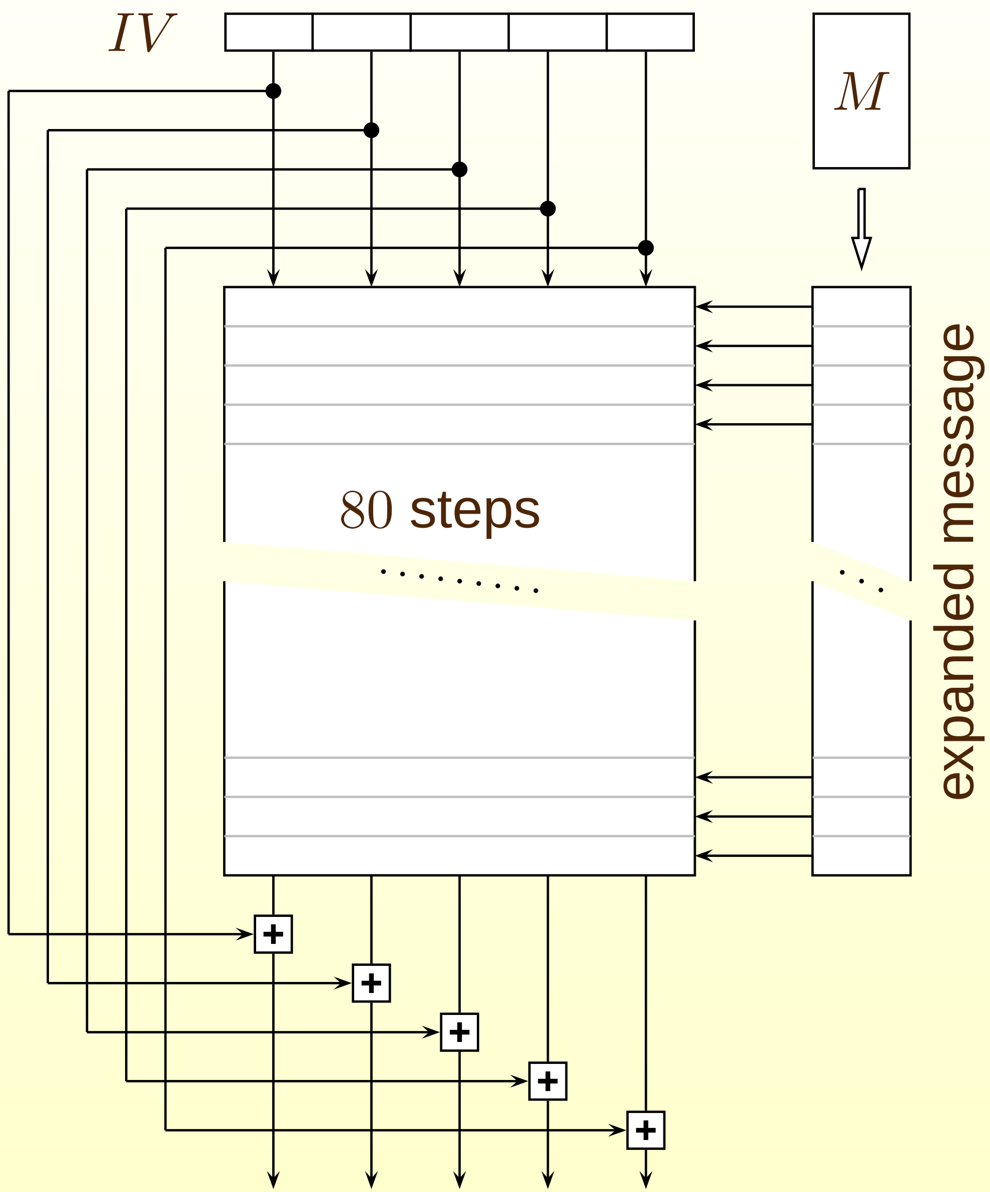
Compression function - function that maps longer inputs to shorter outputs $f : \{0,1\}^{n+k} \rightarrow \{0,1\}^k$.



$$\begin{aligned} h_0 &\leftarrow IV \\ h_i &\leftarrow f(M_{i-1}||h_{i-1}) \\ & \qquad i = 1, \ldots, d \\ h(M) &:= h_d \end{aligned}$$

If the compression function f is secure (one-way and collision-resistant) then the iterative hash function h is also secure.

The structure of SHA : compression function



$$(A_0, B_0, C_0, D_0, E_0) \leftarrow IV.$$

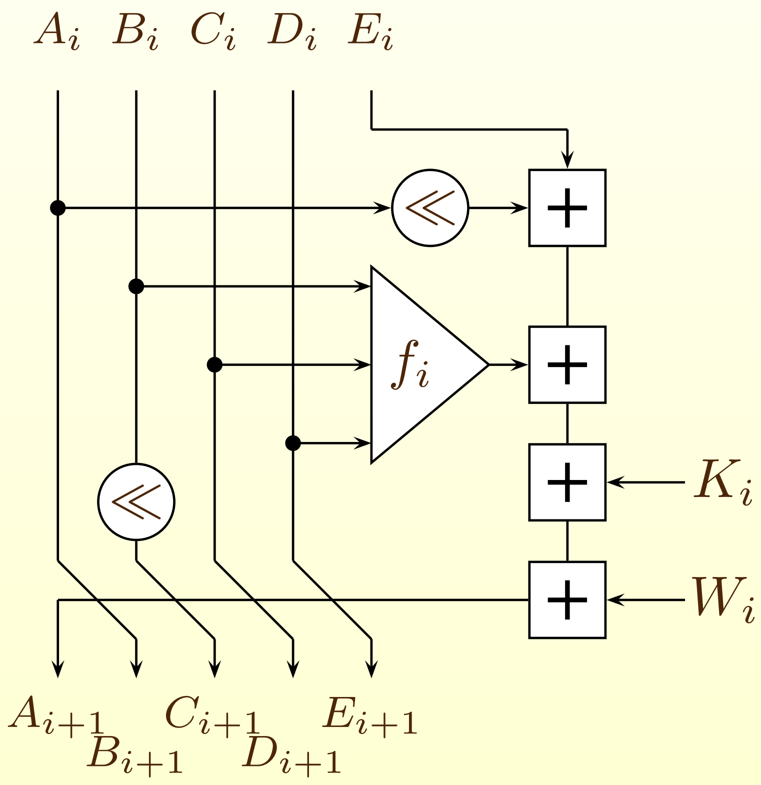
80 steps of the form

$$(A_i, B_i, C_i, D_i, E_i)$$
$$\downarrow$$
$$(A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1})$$

Finally,

$$f(M, IV) = A_0 \boxplus A_{80} || B_0 \boxplus B_{80} ||$$
$$C_0 \boxplus C_{80} || D_0 \boxplus D_{80} ||$$
$$E_0 \boxplus E_{80}$$

The structure of SHA : step transformation



$$\begin{aligned}
A_{i+1} &= E_i \boxplus ROL^5(A_i) \boxplus f_i(B_i, C_i, D_i) \boxplus \\
&\qquad W_i \boxplus K_i, \\
B_{i+1} &= A_i, \\
C_{i+1} &= ROL^{30}(B_i), \\
D_{i+1} &= C_i, \\
E_{i+1} &= D_i, \qquad\qquad i = 0, \ldots, 79
\end{aligned}$$

$$f_i(B, C, D) =$$

$$\begin{cases} BC \vee (\neg B)D & \text{for } 0 \le i \le 19 \\ B \oplus C \oplus D & \text{for } 20 \le i \le 39 \\ BC \vee BD \vee CD & \text{for } 40 \le i \le 59 \\ B \oplus C \oplus D & \text{for } 60 \le i \le 79 \end{cases}$$

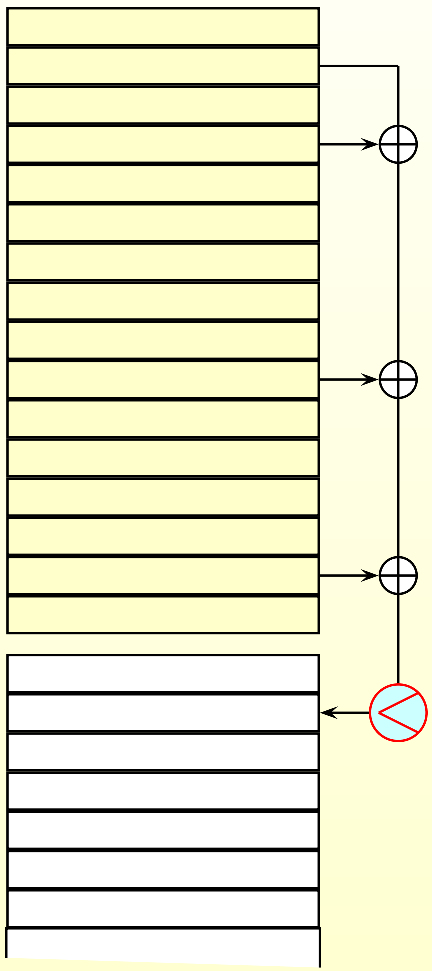
The structure of SHA : message expansion process

For SHA-0:

$$W_i = \begin{cases} M_i & \text{for } 0 \leq i \leq 15, \\ W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16} & \text{for } 16 \leq i \leq 79, \end{cases}$$

For SHA-1:

$$W_i = \begin{cases} M_i & \text{for } 0 \leq i \leq 15, \\ ROL^1(W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) & \text{for } 16 \leq i \leq 79, \end{cases}$$



- Note that the operation is linear in respect of $\oplus$ operation, so $E_1(M \oplus M') = E_1(M) \oplus E_1(M')$.
- SHA-1 differs from SHA-0 only by the rotation in the message expansion.

Differential attack on hash functions

Differential attacks are used for finding collisions.

Idea: Find a difference Δ such that

$$h(M) = h(M \oplus \Delta)$$

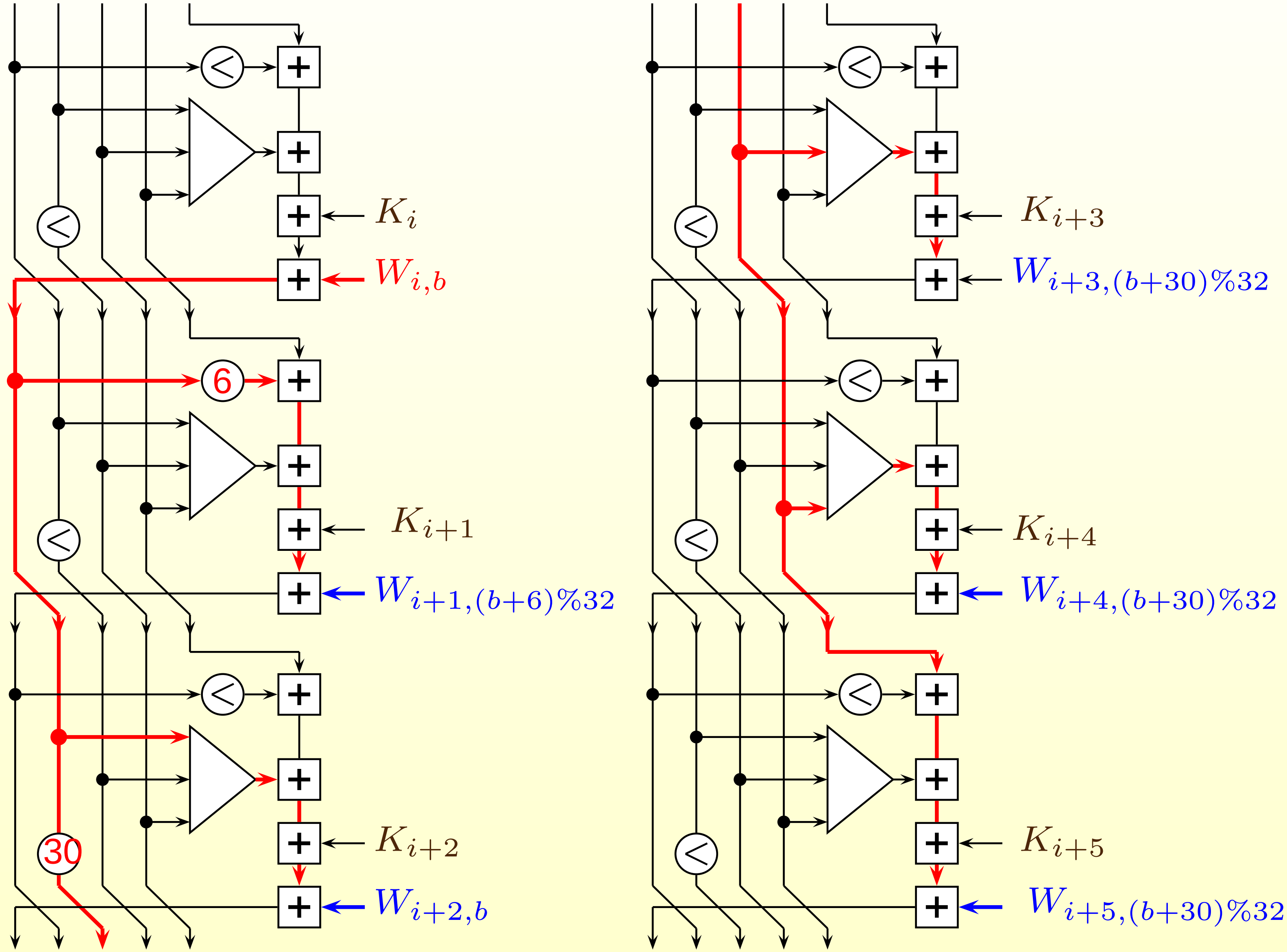
and we know:

- how to construct M,

or

- that we can find a suitable M among random messages with probability higher than $2^{-\text{hash length}/2}$ (faster than generic birthday attack)

Differential attack : disturbance – corrections



Differential attack : probability of success (1) : addition

Disturbance - corrections strategy works if all additions and Boolean functions f_i behave like linear operations in respect of $\oplus$.

Differential attack : probability of success (1) : addition

Disturbance - corrections strategy works if all additions and Boolean functions f_i behave like linear operations in respect of $\oplus$.

This is true for addition when flip of the bit in the message does not generate carries.

Differential attack : probability of success (1) : addition

Disturbance - corrections strategy works if all additions and Boolean functions f_i behave like linear operations in respect of $\oplus$.

This is true for addition when flip of the bit in the message does not generate carries.

Denote $Z_i = E_i \boxplus ROL^5(A_i) \boxplus f_i(B_i, C_i, D_i) \boxplus K_i$.

Z_i (other) :	...	1	0	0	1	0	1
W_i (mesg) :	...	1	1	$0 \to 1$	0	0	1
A_{i+1} (sum):	...	0	1	$0 \to 1$	1	1	0

Z_i (other) :	...	1	0	1	1	0	1
W_i (mesg) :	...	1	1	$0 \to 1$	0	0	1
A_{i+1} (sum):	...	1	0	$1 \to 0$	1	1	0

Differential attack : probability of success (1) : addition

Disturbance - corrections strategy works if all additions and Boolean functions f_i behave like linear operations in respect of $\oplus$.

This is true for addition when flip of the bit in the message does not generate carries.

Denote $Z_i = E_i \boxplus ROL^5(A_i) \boxplus f_i(B_i, C_i, D_i) \boxplus K_i$.

Z_i (other) :	...	1	0	0	1	0	1
W_i (mesg) :	...	1	1	$0 \to 1$	0	0	1
A_{i+1} (sum):	...	0	1	$0 \to 1$	1	1	0

Z_i (other) :	...	1	0	1	1	0	1
W_i (mesg) :	...	1	1	$0 \to 1$	0	0	1
A_{i+1} (sum):	...	1	0	$1 \to 0$	1	1	0

Every bit (except for the most significant ones) adds a factor $1/2$.

Differential attack : probability of success (2) : Boolean functions

Let $\delta f = f(x, y, z) \oplus f(x \oplus \delta_x, y \oplus \delta_y, x \oplus \delta_z)$,
$f_{if}(x, y, z) = xy \vee (\neg x)z = xy \oplus xz \oplus z$,
$f_{maj}(x, y, z) = xy \oplus xz \oplus yz$.

differences				conditions to behave like XOR, i.e. $\delta f = \delta f_{xor}$			
δ_x	δ_y	δ_z	δf_{xor}	f_{if}	Prob	f_{maj}	Prob.
1	0	0	1	$y \oplus z = 1$	1/2	$y \oplus z = 1$	1/2
0	1	0	1	$x = 1$	1/2	$x \oplus z = 1$	1/2
0	0	1	1	$x = 0$	1/2	$x \oplus y = 1$	1/2
1	1	0	0	$x \oplus y \oplus z = 1$	1/2	$x \oplus y = 1$	1/2
1	0	1	0	$x \oplus y \oplus z = 1$	1/2	$x \oplus z = 1$	1/2
0	1	1	0	never	0	$y \oplus z = 1$	1/2
1	1	1	1	$y \oplus z = 1$	1/2	always	1

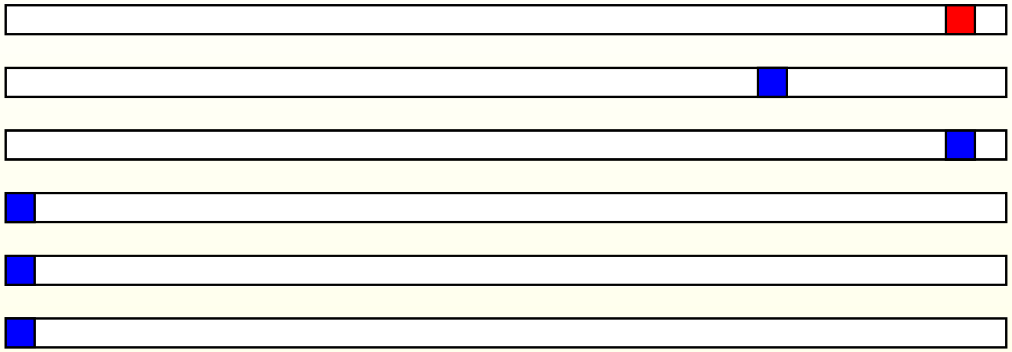
Differential attack : probability of success (2) : Boolean functions

Let $\delta f = f(x, y, z) \oplus f(x \oplus \delta_x, y \oplus \delta_y, x \oplus \delta_z)$,
$f_{if}(x, y, z) = xy \vee (\neg x)z = xy \oplus xz \oplus z$,
$f_{maj}(x, y, z) = xy \oplus xz \oplus yz$.

differences				conditions to behave like XOR, i.e. $\delta f = \delta f_{xor}$			
δ_x	δ_y	δ_z	δf_{xor}	f_{if}	Prob	f_{maj}	Prob.
1	0	0	1	$y \oplus z = 1$	1/2	$y \oplus z = 1$	1/2
0	1	0	1	$x = 1$	1/2	$x \oplus z = 1$	1/2
0	0	1	1	$x = 0$	1/2	$x \oplus y = 1$	1/2
1	1	0	0	$x \oplus y \oplus z = 1$	1/2	$x \oplus y = 1$	1/2
1	0	1	0	$x \oplus y \oplus z = 1$	1/2	$x \oplus z = 1$	1/2
0	1	1	0	never	0	$y \oplus z = 1$	1/2
1	1	1	1	$y \oplus z = 1$	1/2	always	1

Every Boolean function different from XOR adds a factor $1/2$ and we cannot have two adjacent changes in first 16 steps

Differential attack : from disturbance pattern to full differential



Let d denotes the pattern of disturbance bits. Then the complete differential pattern can be obtained as

$$\begin{aligned}\Delta = d \oplus\ & Delay^1(ROL^6(d)) \oplus \\ & Delay^2(d) \oplus \\ & Delay^3(ROL^{30}(d)) \oplus \\ & Delay^4(ROL^{30}(d)) \oplus \\ & Delay^5(ROL^{30}(d)),\end{aligned}$$

where $Delay^k(W)$ means inserting k zero words before W and discarding the last k words of W.

Conditions for the disturbance pattern

In order to construct difference pattern Δ (disturbance + corrections) from a disturbance pattern d, d has to satisfy the following conditions:

Conditions for the disturbance pattern

In order to construct difference pattern Δ (disturbance + corrections) from a disturbance pattern d, d has to satisfy the following conditions:

- d has to be the result of the expansion operation,

Conditions for the disturbance pattern

In order to construct difference pattern Δ (disturbance + corrections) from a disturbance pattern d, d has to satisfy the following conditions:

- d has to be the result of the expansion operation,
- d has to end with five zero words (because each disturbance is corrected in the next 5 steps, so no disturbance may occur after the word 74),

Conditions for the disturbance pattern

In order to construct difference pattern Δ (disturbance + corrections) from a disturbance pattern d, d has to satisfy the following conditions:

- d has to be the result of the expansion operation,
- d has to end with five zero words (because each disturbance is corrected in the next 5 steps, so no disturbance may occur after the word 74),
- after delaying d by up to 5 words the delayed patterns $Delay^1(d), \ldots, Delay^5(d)$ must also be the result of the expansion of theirs first 16 words,

Conditions for the disturbance pattern

In order to construct difference pattern Δ (disturbance + corrections) from a disturbance pattern d, d has to satisfy the following conditions:

- d has to be the result of the expansion operation,
- d has to end with five zero words (because each disturbance is corrected in the next 5 steps, so no disturbance may occur after the word 74),
- after delaying d by up to 5 words the delayed patterns $Delay^1(d), \ldots, Delay^5(d)$ must also be the result of the expansion of theirs first 16 words,
- d has both the minimal Hamming weight and the maximal number of non-zero bits in position 1.

The search for disturbance patterns for SHA-0: (1)

$$W_i = \begin{cases} M_i & \text{for } 0 \leq i \leq 15 \\ W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16} & \text{for } 16 \leq i \leq 79, \end{cases}$$

In SHA-0, bits in different positions are independent!

The search for disturbance patterns for SHA-0: (1)

$$W_i = \begin{cases} M_i & \text{for } 0 \leq i \leq 15 \\ W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16} & \text{for } 16 \leq i \leq 79, \end{cases}$$

In SHA-0, bits in different positions are independent!

- changing bits in position k in the message words M_j will affect only bits in position k in expanded message,

The search for disturbance patterns for SHA-0: (1)

$$W_i = \begin{cases} M_i & \text{for } 0 \le i \le 15 \\ W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16} & \text{for } 16 \le i \le 79, \end{cases}$$

In SHA-0, bits in different positions are independent!

- changing bits in position k in the message words M_j will affect only bits in position k in expanded message,
- message expansion process can be seen as 32 independent copies of the expansion of 16 bits to 80 bits using the relation

$$w_i = w_{i-3} \oplus w_{i-8} \oplus w_{i-14} \oplus w_{i-16} \qquad 16 \le i \le 79$$

where $w_i \in \mathbb{F}_2$.

The search for disturbance patterns for SHA-0: (2)

- there are 2^{16} candidates for disturbance patterns,

The search for disturbance patterns for SHA-0: (2)

- there are 2^{16} candidates for disturbance patterns,
- there are 2^{11} patterns such that $Delay^5(d)$ is the result of expansion,

The search for disturbance patterns for SHA-0: (2)

- there are 2^{16} candidates for disturbance patterns,
- there are 2^{11} patterns such that $Delay^5(d)$ is the result of expansion,
- there are 2^6 patterns such that $Delay^5(d)$ is the result of expansion and pattern ends with five zero bits (63 usable patterns, excluding all-zero) Minimal weight is 27.

The search for disturbance patterns for SHA-0: (2)

- there are 2^{16} candidates for disturbance patterns,
- there are 2^{11} patterns such that $Delay^5(d)$ is the result of expansion,
- there are 2^6 patterns such that $Delay^5(d)$ is the result of expansion and pattern ends with five zero bits (63 usable patterns, excluding all-zero) Minimal weight is 27.
- there are only 5 disturbance patterns such that there are no adjacent '1' bits in first 16 bits

0001000000010010 0000001000011011 0111111011010010 0001010100101010 0010111001100000

0010001000000010 1111011000111000 0001010001000100 1001001110110011 0000111110000000

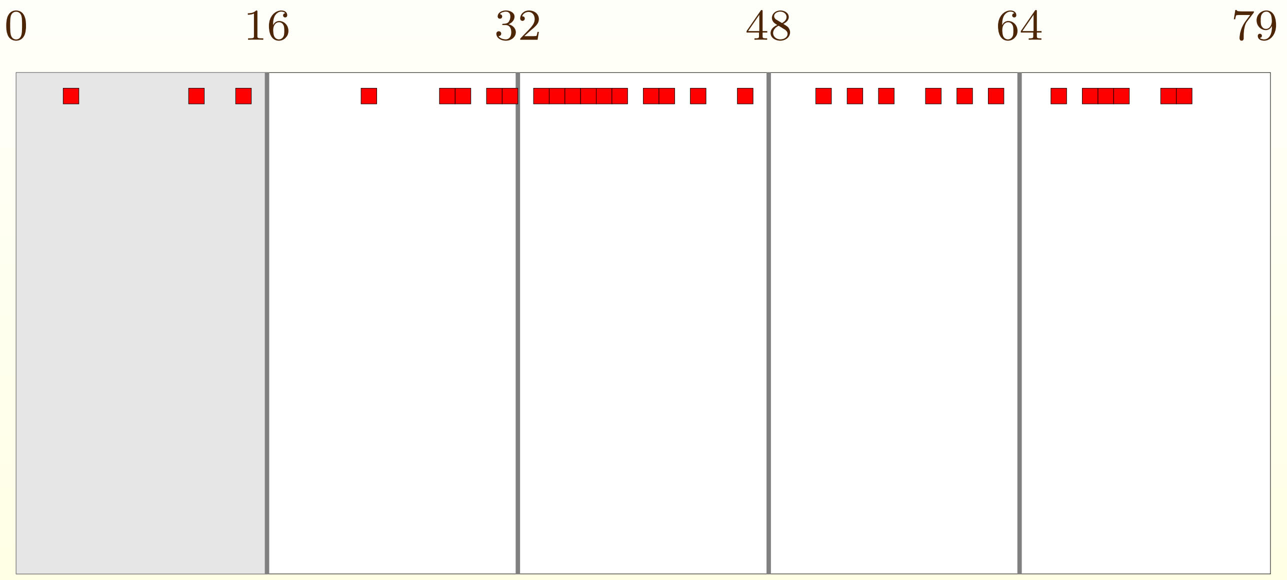
0100001010010001 1110010110000011 1000000000001100 0000110110000001 1000101101100000

0010100101000001 1111001111001100 0111111101101111 0000110001010101 1101001010000000

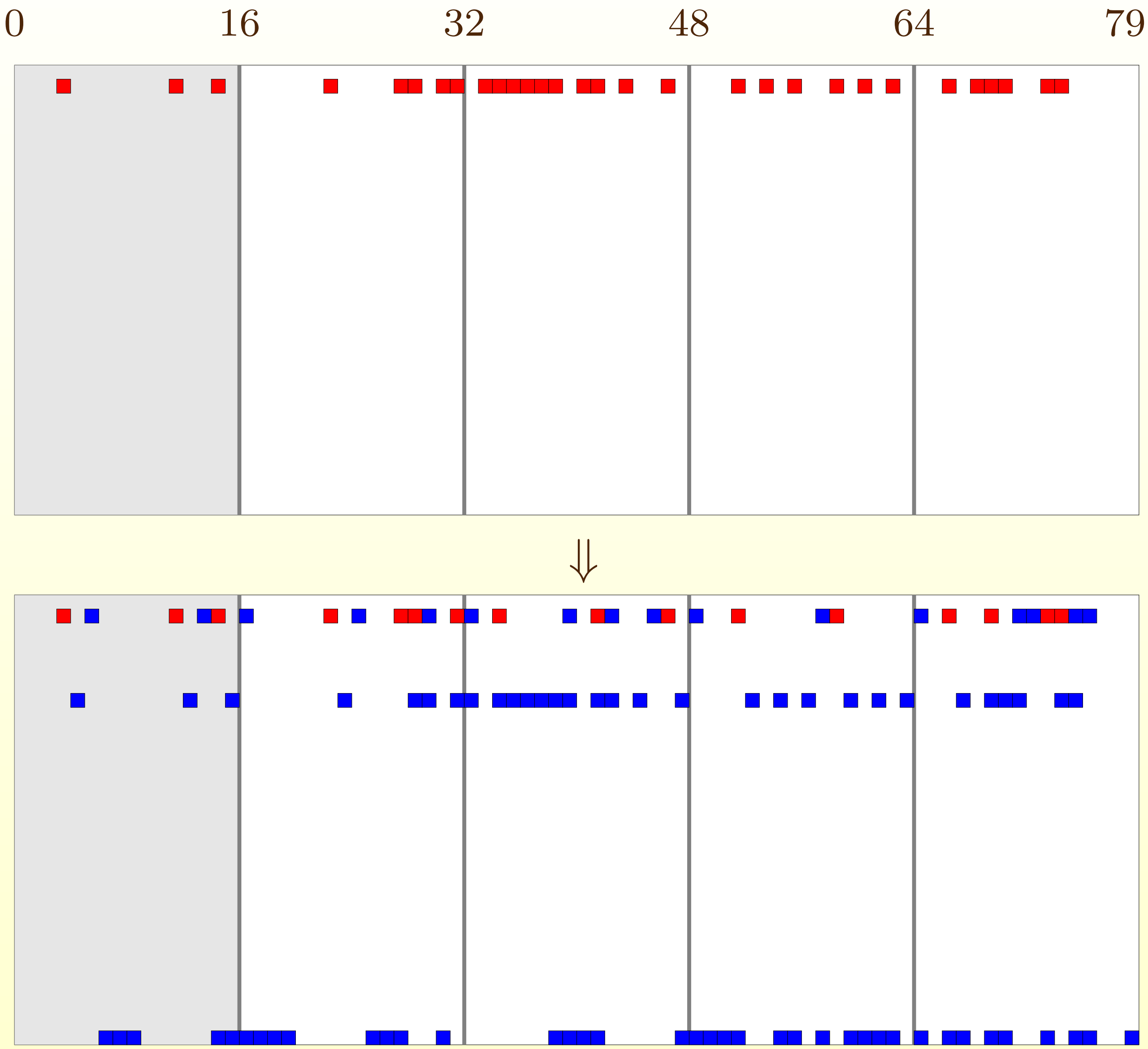
0001010010100000 1111100111100110 0011111110110111 1000011000101010 1110100101000000

with weights: 30, 30, 27, 39, 39.

Pattern for differential attack on SHA-0



Pattern for differential attack on SHA-0



differential pattern with probability 2^{-68}

What about SHA-1 ?

1. The only difference is in the message expansion algorithm, so the idea of *disturbance - corrections* works also for SHA-1 - the round structure is the same
2. how to find disturbance patterns that can give rise to corrective patterns ?

Properties of the message expansion in SHA-1

$$W_i = \begin{cases} M_i & \text{for } 0 \leq i \leq 15 \\ ROL^1(W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) & \text{for } \ 16 \leq i \leq 79, \end{cases}$$

All operations are $\mathbb{F}_2$-linear, so we can describe the whole message expansion process as a linear function

$$E_1 \ : \ \mathbb{F}_2^{512} \longrightarrow \mathbb{F}_2^{2560}$$

The function A producing 16 new words $(W_{i+1}, \ldots, W_{i+16})$ out of 16 old ones $(W_{i-15}, \ldots, W_i)$ using the recurrence formula is a linear bijection of space $\mathbb{F}_2^{512}$,

$$A \ : \ \mathbb{F}_2^{512} \longrightarrow \mathbb{F}_2^{512}.$$

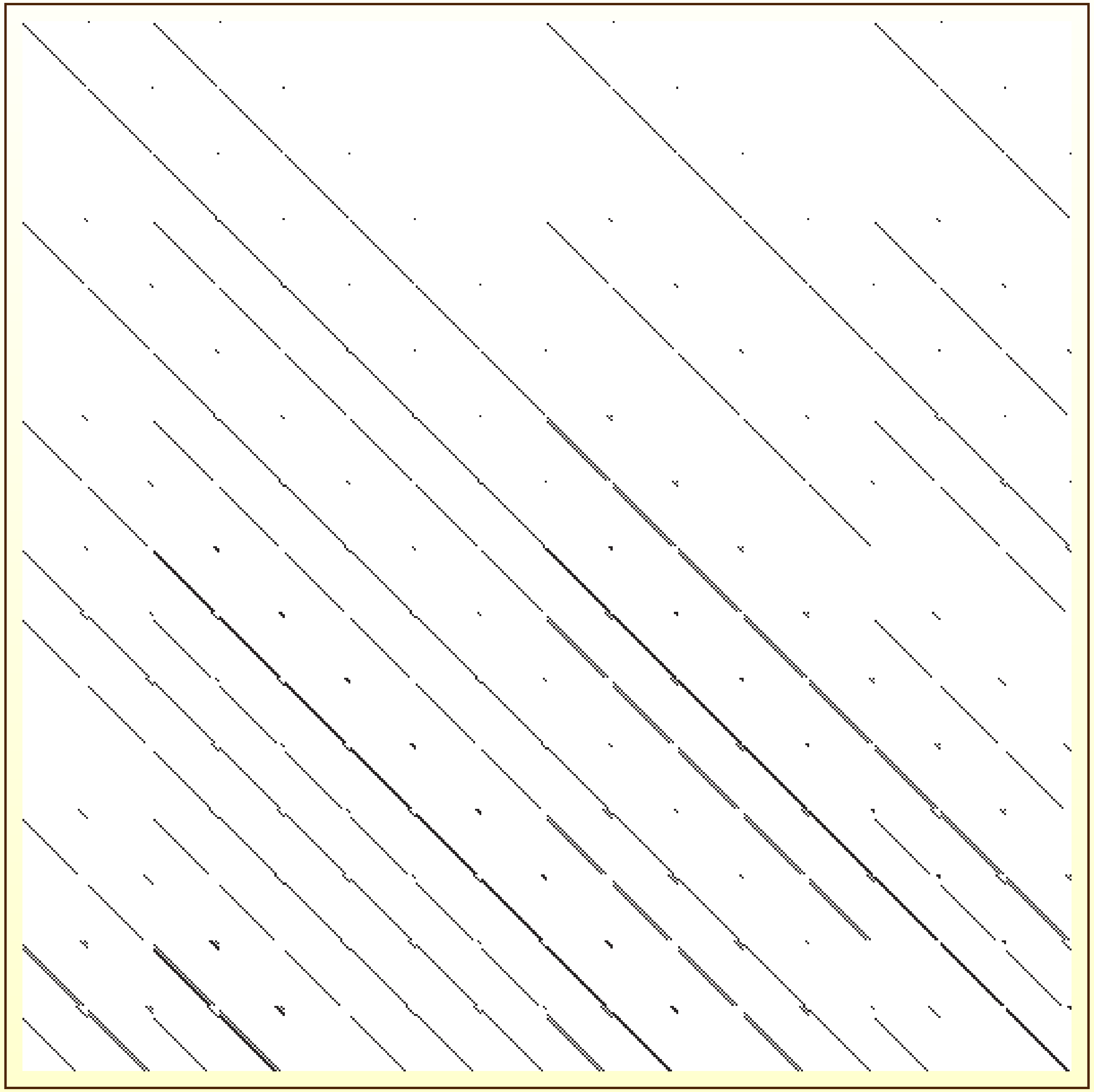
Message expansion process: Relation between A and E_1

If we consider a message as a bit vector $m \in \mathbb{F}_2^{512}$, we can write

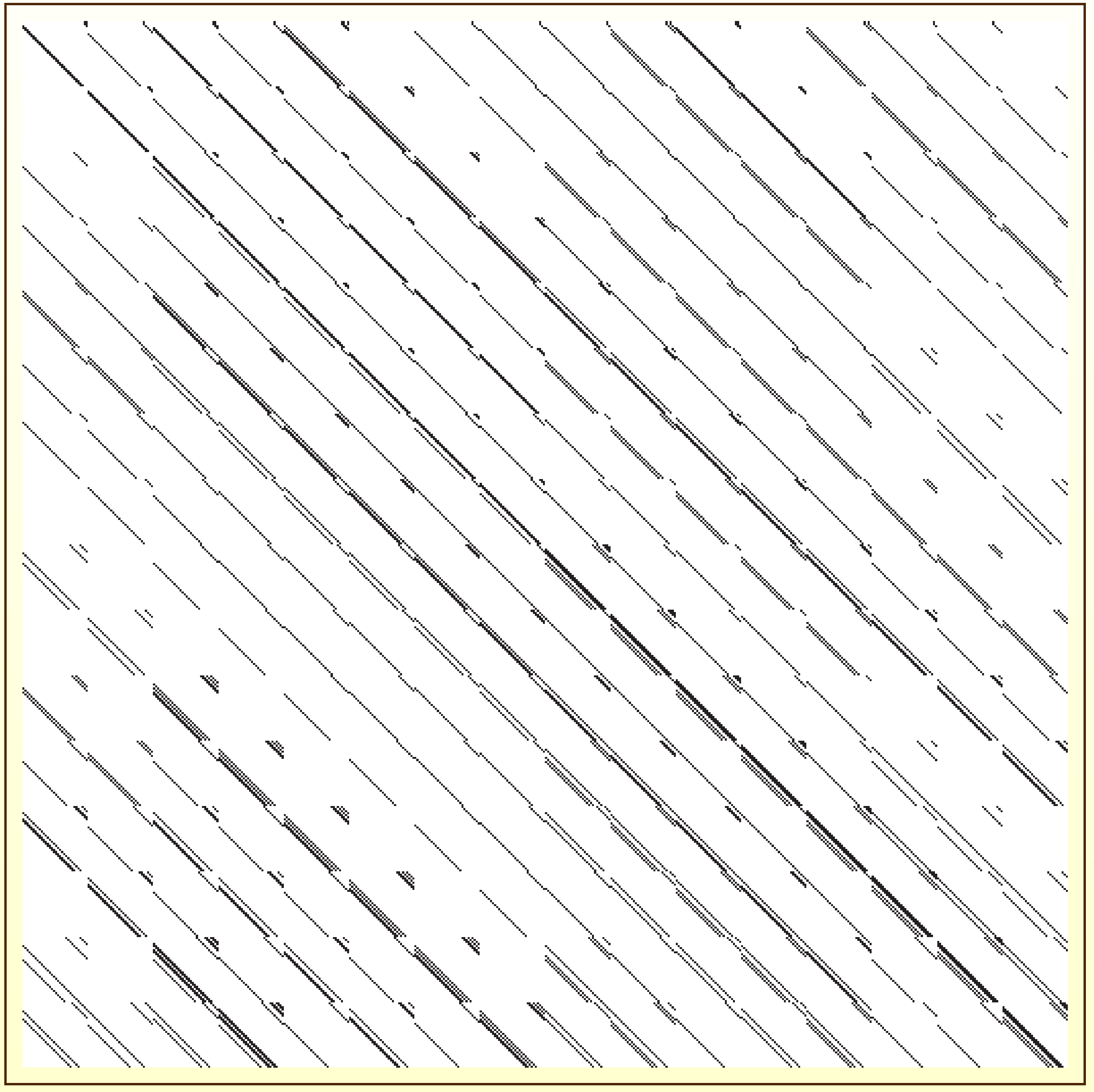
$$E_1(m) = \left[\begin{array}{c} I_{512} \\ \hline A \\ \hline A^2 \\ \hline A^3 \\ \hline A^4 \end{array}\right] \cdot m$$

Denote $L = \left[\begin{array}{c} I_{512} \\ \hline A \\ \hline A^2 \\ \hline A^3 \\ \hline A^4 \end{array}\right]$ for later use.

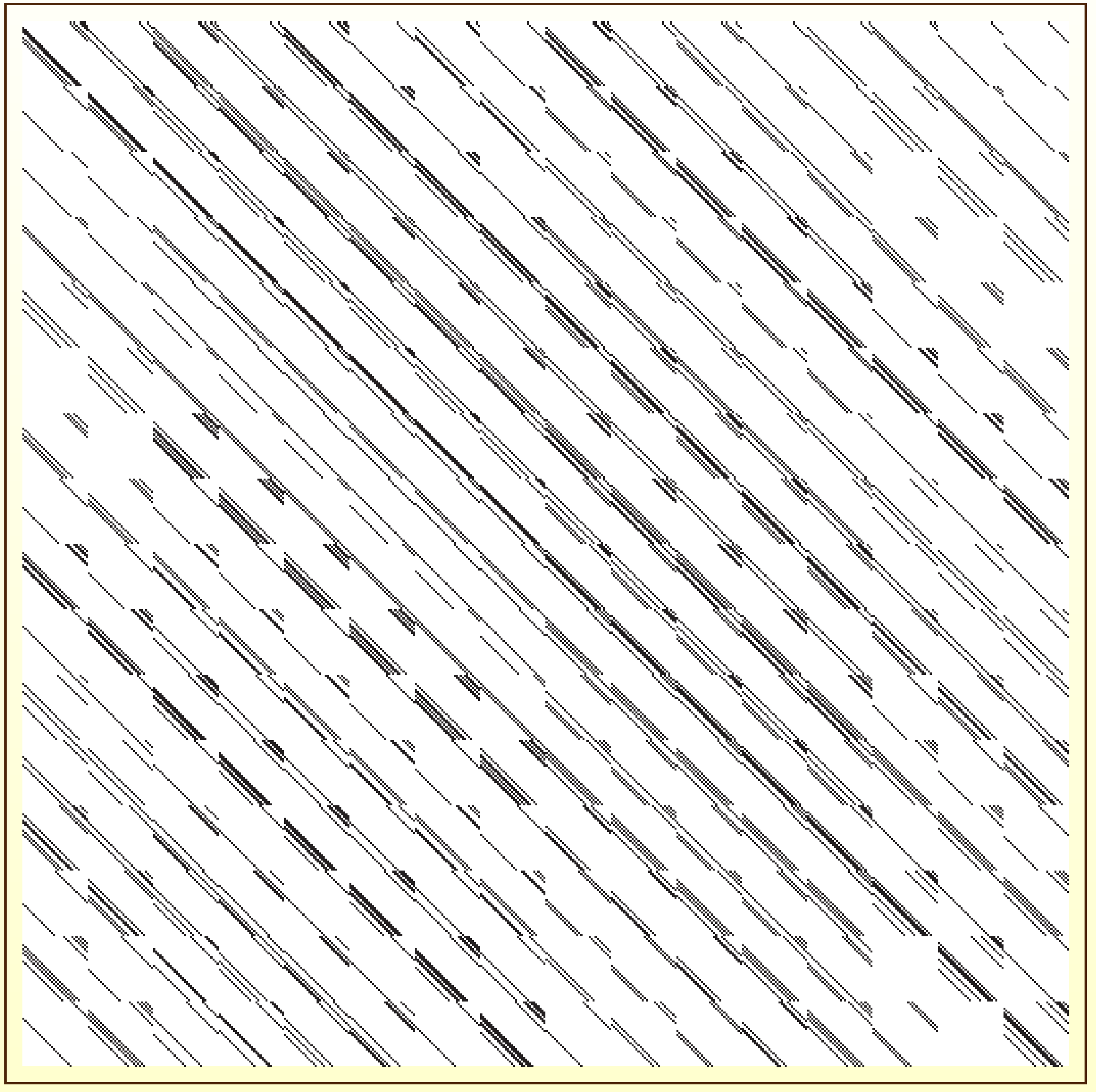
Matrix A



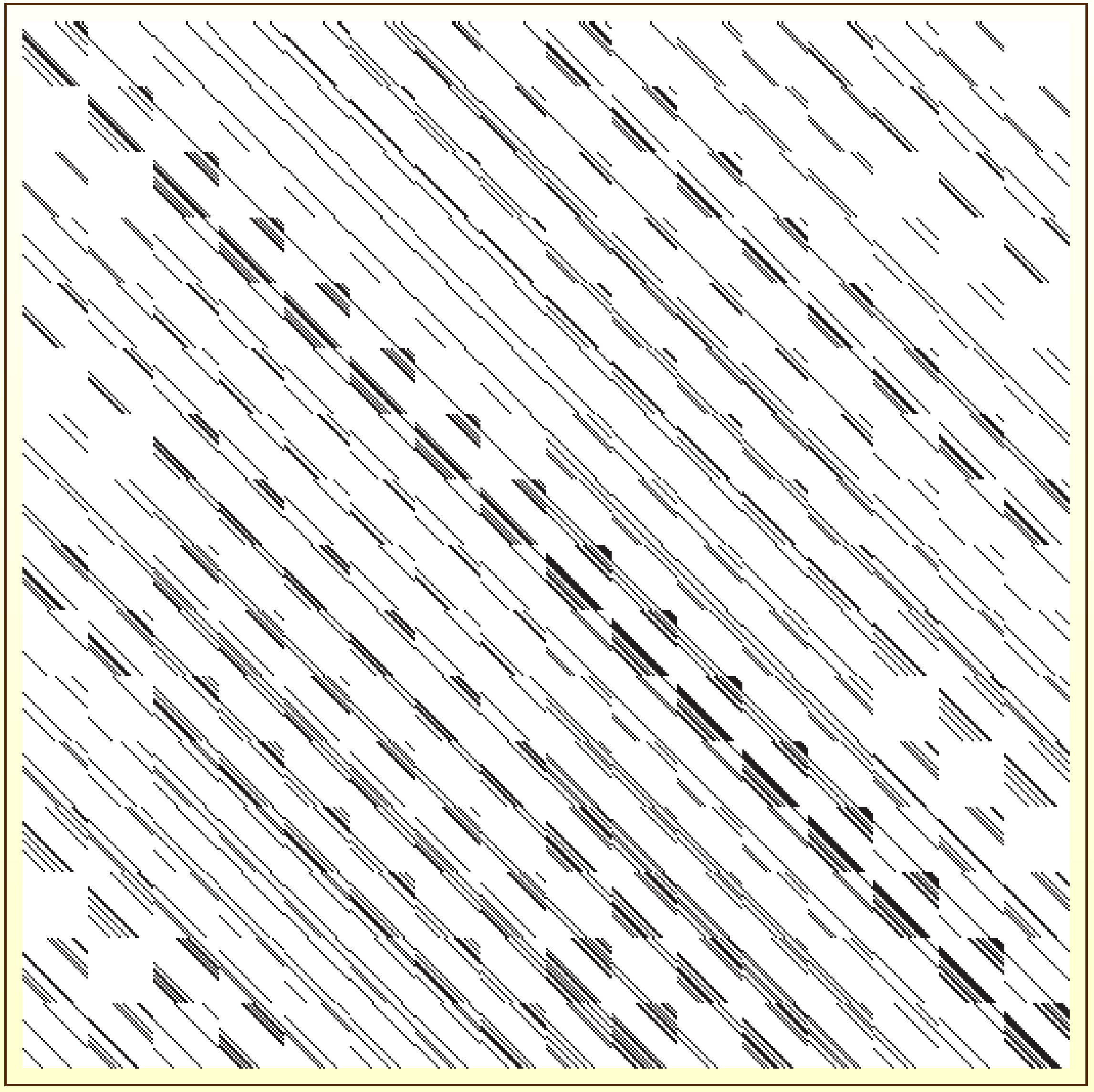
Matrix A^2



Matrix A^3



Matrix A^4



How to find disturbance patterns?

1. d has to be the result of the expansion operation,

$$d = E_1([d_0, \ldots, d_{511}]^T)$$

2. d has to end with five zero words (because each disturbance is corrected in the next 5 steps, so no disturbance may occur after the word 74),

$$d_j = 0, \qquad \text{for } j = 2400, \ldots 2559,$$

3. after delaying d by up to 5 words the delayed patterns $Delay^1(d), \ldots, Delay^5(d)$ must also be the result of the expansion of theirs first 16 words,

$$[\underbrace{0 \ldots 0}_{160 \text{ bits}}\, d_0\ d_1\ \ \ldots\ \ d_{2399}]^T = E_1([0 \ldots 0\ d_0 \ldots\ d_{351}]^T)\ \ .$$

How to find disturbance patterns? (2)

Conditions 1– 3 imply that in fact we are looking for longer bit sequences of 85 words such that

- the first 5 words are zero,
- the next 11 words are chosen in such a way that the rest of the words is the result of the expansion of the first 16, and
- the last 5 words are zero again.

If we denote first 5 words with indices $-5, -4, \ldots, -1$ words $0, \ldots 79$ are the words of a disturbance pattern.

In matrix notation: we are looking for bit vectors $m \in \mathbb{F}_2^{512}$ such that

- $A^4 \cdot m$ has $5 \cdot 32 = 160$ trailing zero bits,
- $A^{-1} \cdot m$ has $5 \cdot 32 = 160$ trailing zero bits.

How to find disturbance patterns? (3)

$$\left[\begin{array}{c} * \\ \vdots \\ * \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{cccc} \alpha_{0,0} & \cdots & \cdots & \alpha_{0,511} \\ \vdots & & & \vdots \\ \alpha_{351,0} & & & \alpha_{351,511} \\ \hline \alpha_{352,0} & \cdots & \cdots & \alpha_{352,511} \\ \vdots & & & \vdots \\ \alpha_{511,0} & \cdots & \cdots & \alpha_{511,511} \end{array}\right] \cdot \left[\begin{array}{c} m_0 \\ \vdots \\ m_{351} \\ m_{352} \\ \vdots \\ m_{511} \end{array}\right] = A^{-1} \cdot m$$

$$\left[\begin{array}{c} * \\ \vdots \\ * \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] = \left[\begin{array}{cccc} \hat{a}_{0,0} & \cdots & \cdots & \hat{a}_{0,511} \\ \vdots & & & \vdots \\ \hat{a}_{351,0} & & & \hat{a}_{351,511} \\ \hline \hat{a}_{352,0} & \cdots & \cdots & \hat{a}_{352,511} \\ \vdots & & & \vdots \\ \hat{a}_{511,0} & \cdots & \cdots & \hat{a}_{511,511} \end{array}\right] \cdot \left[\begin{array}{c} m_0 \\ \vdots \\ m_{351} \\ m_{352} \\ \vdots \\ m_{511} \end{array}\right] = A^4 \cdot m$$

How to find disturbance patterns? (4)

We are looking for patterns $d \in \mathbb{F}_2^{512}$ such that

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = \left[\begin{array}{cccc} \alpha_{352,0} & \cdots & \cdots & \alpha_{352,511} \\ \vdots & & & \vdots \\ \alpha_{511,0} & \cdots & \cdots & \alpha_{511,511} \\ \hline \hat{a}_{352,0} & \cdots & \cdots & \hat{a}_{352,511} \\ \vdots & & & \vdots \\ \hat{a}_{511,0} & \cdots & \cdots & \hat{a}_{511,511} \end{array}\right] \cdot \begin{bmatrix} m_0 \\ \vdots \\ m_{351} \\ m_{352} \\ \vdots \\ m_{511} \end{bmatrix}$$

Notation: $A[p :: q]$ - matrix created by taking rows of the matrix A from p-th row to q-th row. Then the equation above can be written as

$$0 = \Psi \cdot m \qquad \text{where}$$

$$\Psi = \left[\begin{array}{c} A^{-1}[352 :: 511] \\ \hline A^{4}[352 :: 511] \end{array}\right]$$

How to find disturbance patterns? (5)

Thus, all disturbance patterns $d \in \mathbb{F}_2^{2560}$ we are looking for are created as expansions of bit vectors $m \in \mathbb{F}_2^{512}$ from the linear subspace

$$\ker \Psi.$$

Experimentally we have found that

$$\dim \ker \Psi = 192.$$

This shows that the set of all disturbance patterns for *disturbance - corrections* technique constitutes a linear code $\mathcal{C}$ of length 2560 and dimension 192.

Disturbance patterns for reduced variants of SHA-1

If we want to look for patterns suitable for SHA-1 reduced to only s steps, we need to take a different matrix Ψ:

$$\Psi_s = \left[\frac{A^{-1}[352 :: 511]}{L[32(s-4) :: 32s + 31]} \right]$$

where L is the matrix of the full expansion process E_1.

How to find good patterns?

Finding the best pattern is equivalent to finding the minimal weight codeword in $\mathcal{C}$.

This problem is NP-hard in general .....

However, it looks like this code is quite particular and we were able to achieve good results.

Algorithm: modification of [J.S. Leon , F. Chabaud]

- permute the columns of the generating matrix randomly
- apply a gaussian elimination on the rows of the matrix to get

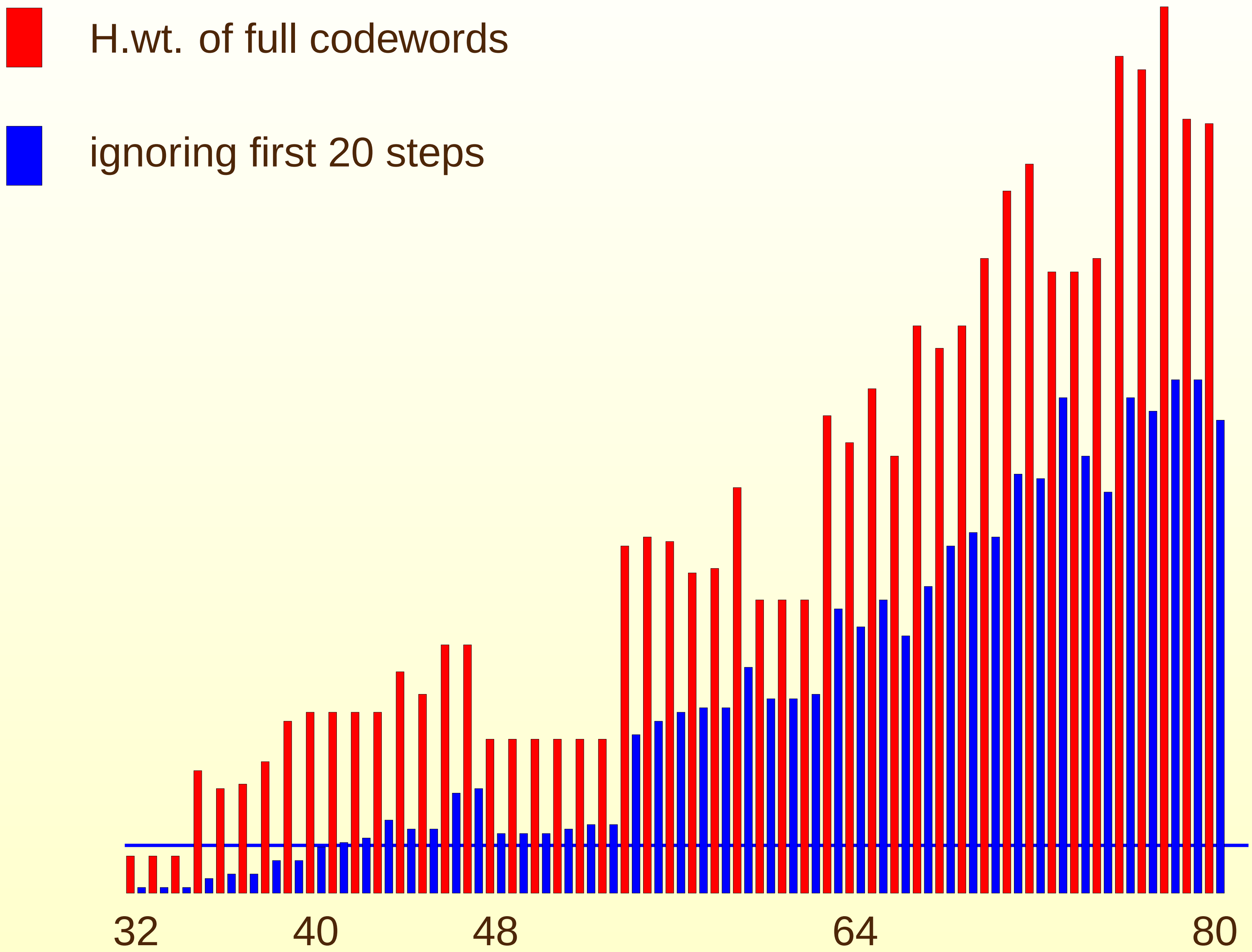
$$G = (\ I \mid Z\ )$$

- search for combinations of up to p rows of Z that lead to codewords with small weight

Results

steps	wt	wt_{20+}	steps	wt	wt_{20+}	steps	wt	wt_{20+}
32	9	2	50	35	14	68	> 122	> 78
33	9	2	51	35	15	69	> 127	> 81
34	9	2	52	35	16	70	> 142	> 80
35	28	4	53	35	16	71	> 157	> 94
36	24	5	54	78	36	72	> 163	> 93
37	25	5	55	80	39*	73	> 139	> 111
38	30	8	56	79	41	74	> 139	> 98
39	39	8*	57	72	42	75	> 142	> 90
40	41	11	58	73	42	76	> 187	> 111
41	41	12	59	91	51	77	> 184	> 108
42	41	13	60	66	44	78	> 198	> 115
43	41	17	61	66	44	79	> 173	> 115
44	50	15	62	66	45	80	> 172	> 106
45	45	15	63	107	64	81	> 255	> 117
46	56	23	64	> 101	> 60	82	> 242	> 142
47	56	24*	65	> 113	> 66	83	> 215	> 163
48	35	14	66	> 98	> 58	84	> 161	> 101
49	35	14	67	> 127	> 69	85	> 340	> 177

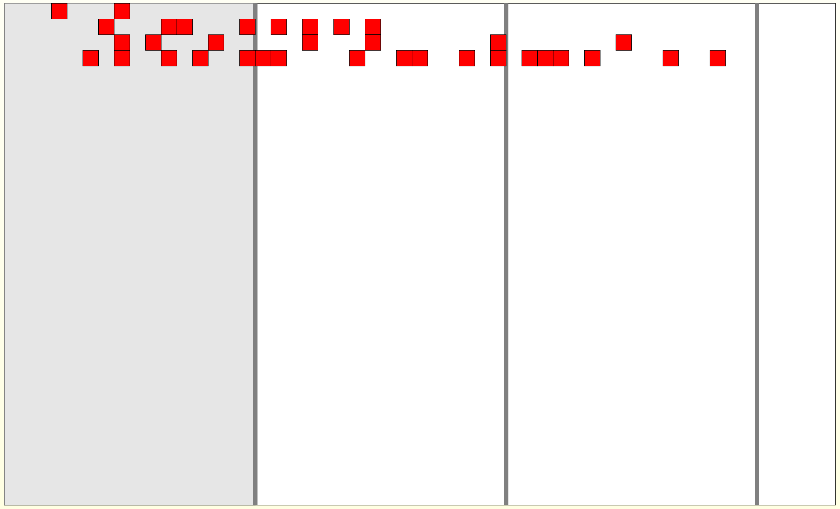
Results: best Hamming weights for different lengths



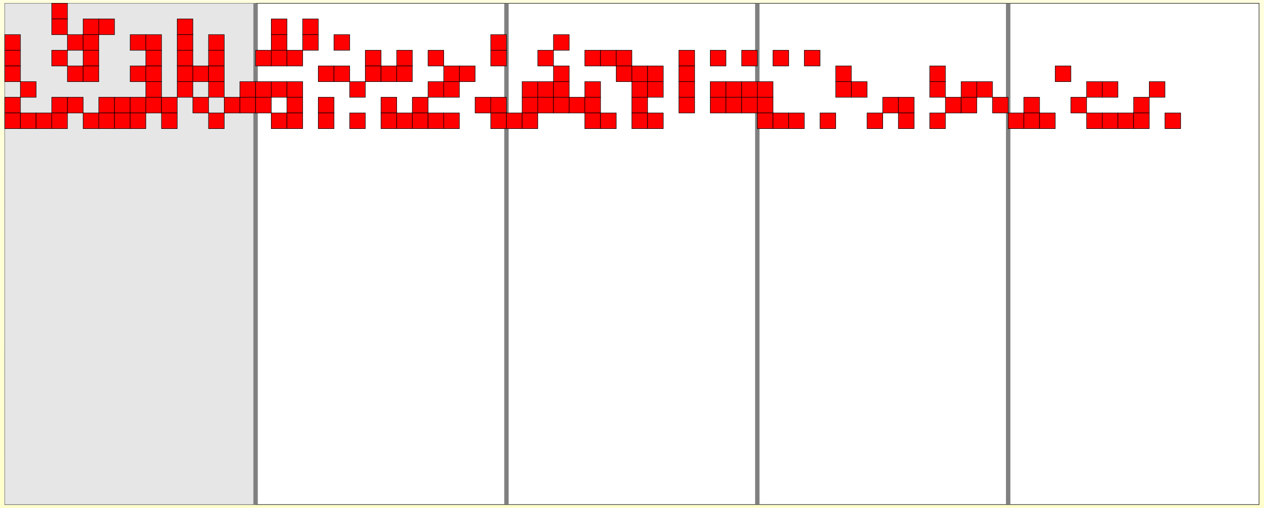
A few examples of patterns

34 steps $(9/2)$

53 steps $(35/16)$



80 steps $(172/120)$



(wt/wt_{20+})

Message expansion backward

The message expansion can be applied "from the end" and then has the form

$$W_i = W_{i+2} \oplus W_{i+8} \oplus W_{i+13} \oplus ROR^1(W_{i+16}), \quad 0 \leq i < 64,$$

where the last 16 words W_{64},...,W_{79} are fixed.

Message expansion backward

The message expansion can be applied "from the end" and then has the form

$$W_i = W_{i+2} \oplus W_{i+8} \oplus W_{i+13} \oplus ROR^1(W_{i+16}), \quad 0 \leq i < 64,$$

where the last 16 words W_{64},...,W_{79} are fixed.

- rotation is applied to only one word distant by 16 steps !

Message expansion backward

The message expansion can be applied “from the end” and then has the form

$$W_i = W_{i+2} \oplus W_{i+8} \oplus W_{i+13} \oplus ROR^1(W_{i+16}), \quad 0 \leq i < 64,$$

where the last 16 words $W_{64}, \ldots, W_{79}$ are fixed.

- rotation is applied to only one word distant by 16 steps !
- much worse avalanche effect

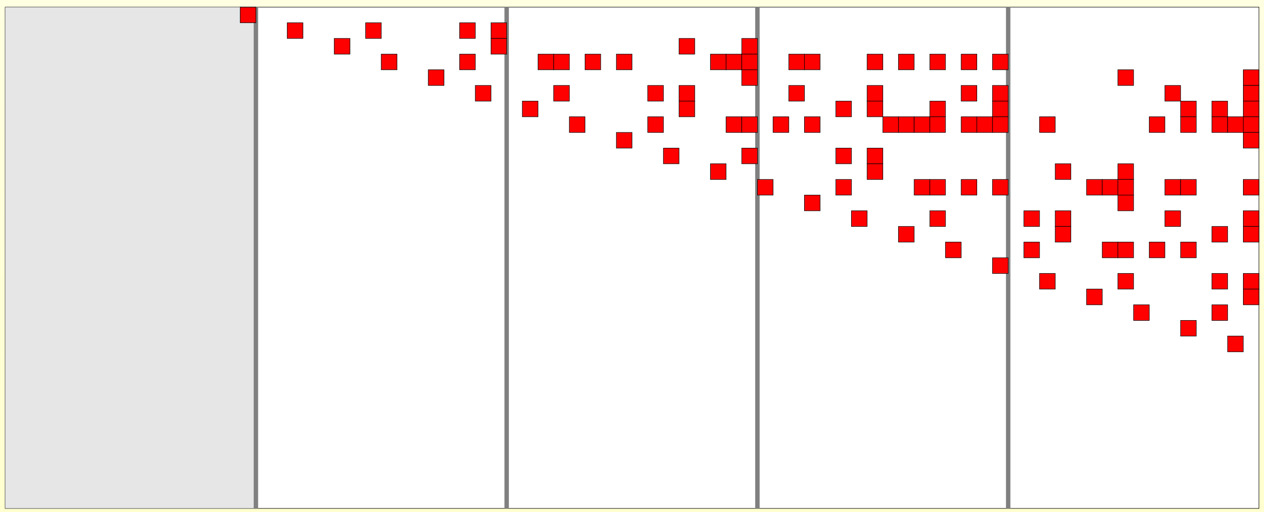
Message expansion backward

The message expansion can be applied “from the end” and then has the form

$$W_i = W_{i+2} \oplus W_{i+8} \oplus W_{i+13} \oplus ROR^1(W_{i+16}), \quad 0 \le i < 64,$$

where the last 16 words $W_{64}, \ldots, W_{79}$ are fixed.

- rotation is applied to only one word distant by 16 steps !
- much worse avalanche effect



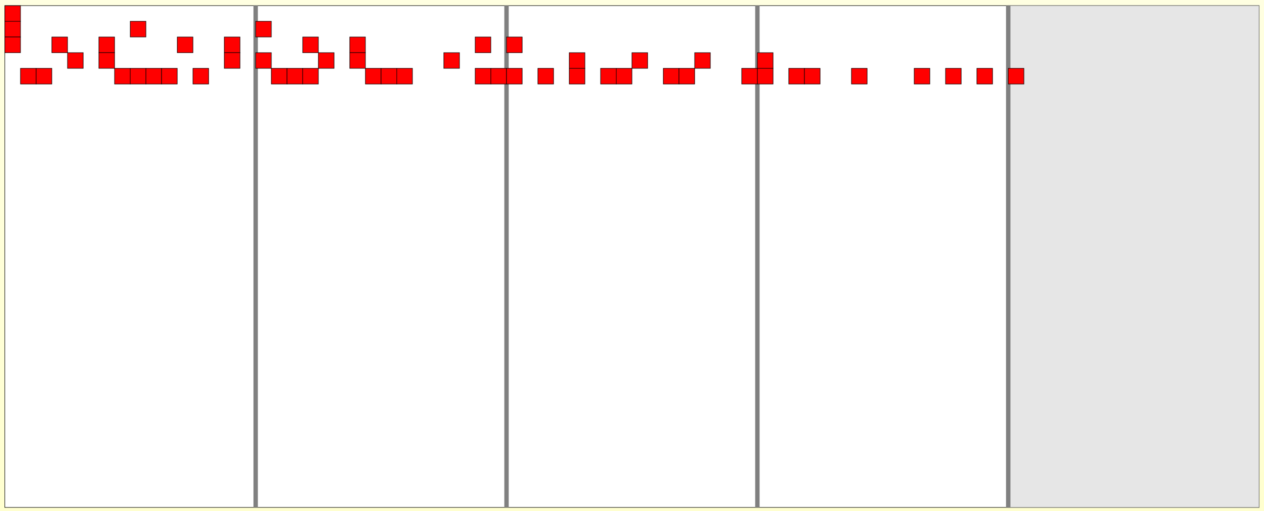
Message expansion backward

The message expansion can be applied “from the end” and then has the form

$$W_i = W_{i+2} \oplus W_{i+8} \oplus W_{i+13} \oplus ROR^1(W_{i+16}), \quad 0 \leq i < 64,$$

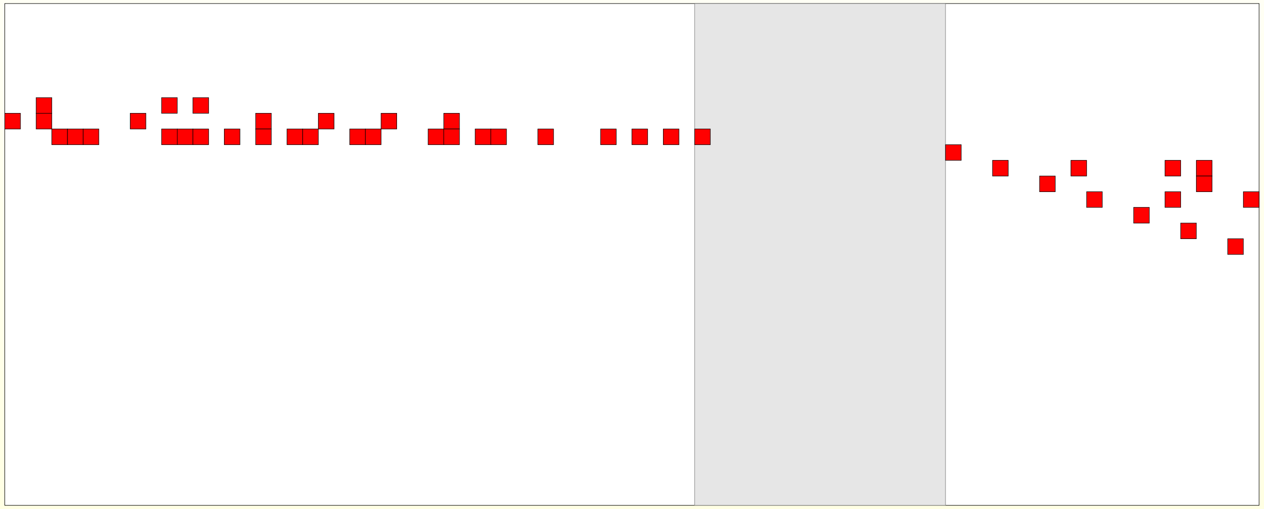
where the last 16 words $W_{64}, \ldots, W_{79}$ are fixed.

- rotation is applied to only one word distant by 16 steps !
- much worse avalanche effect



Minimum weight unrestricted pattern

Minimum weight of the expanded message we could find: 44.



- found in the following way: change one bit in word 44 and expand the segment $44 – 60$ backward-forward
- independently found as a candidate for the minimal weight codeword in unrestricted code

Bounds on the weight of short patterns

To estimate weight of a differential pattern we can divide it into two groups:

- S_1 – set of bits in the same position as the last nonzero bit
- S_2 – set of bits in other positions (right from the initial position)

Bounds on the weight of short patterns

To estimate weight of a differential pattern we can divide it into two groups:

- S_1 – set of bits in the same position as the last nonzero bit
- S_2 – set of bits in other positions (right from the initial position)

It's easy to estimate the size of S_1 : bits in the same position are generated by the recurrence formula

$$w_i = w_{i+2} \oplus w_{i+8} \oplus w_{i+13}$$

Minimal weights of such sequences can be easily found (only 2^{16} possibilities).

Bounds on the weight of short patterns

To estimate weight of a differential pattern we can divide it into two groups:

- S_1 – set of bits in the same position as the last nonzero bit
- S_2 – set of bits in other positions (right from the initial position)

It's easy to estimate the size of S_1 : bits in the same position are generated by the recurrence formula

$$w_i = w_{i+2} \oplus w_{i+8} \oplus w_{i+13}$$

Minimal weights of such sequences can be easily found (only 2^{16} possibilities).

We can't say much about the size of the second set, only that $|S_2| \geq 1$ for patterns longer than 16.

Bounds : 34-step pattern is optimal

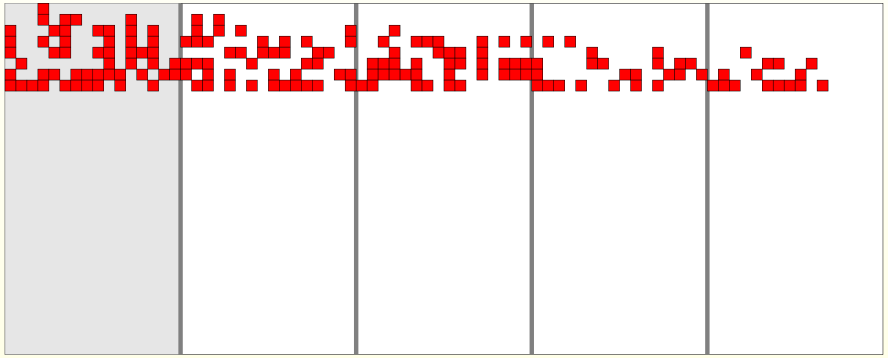
steps	32–34	35–38	39,40	41	42,43	44–47	48,49	50	51
min. wt	8	9	11	13	11	14	16	17	16

steps	52,53	54–56	57–64	65–67	68–71	72	73–75	76,77	78–85
min. wt	17	18	19	23	22	26	24	29	30

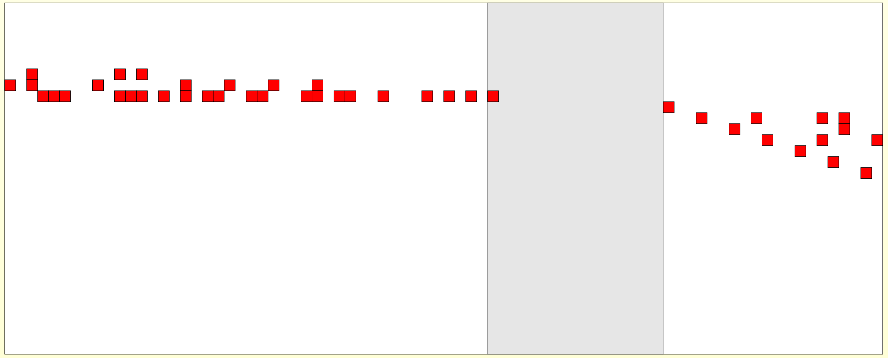
34 steps case: minimal size of S_1 - 8, minimal size of S_2 - 1.
Actual weight : 9

Future work

- Can we use



- Can we use



- Can we construct complete differences in a different way than using *disturbance-corrections* strategy?

The End

Thank you!