

# Differential Fault Analysis on the SHA1 Compression Function

Ludger Hemme and Lars Hoffmann

Giesecke & Devrient GmbH  
Munich

FDTC 2011 - 28. 09. 2011

# Citation bug

---

- Due to Bibtex-error please correct:

[20] **Ruilin Li, Chao Li and Chunye Gong.** Differential Fault Analysis on SHACAL-1. In *FDTC 2009*, pages 120-126. IEEE Computer Society, 2009.

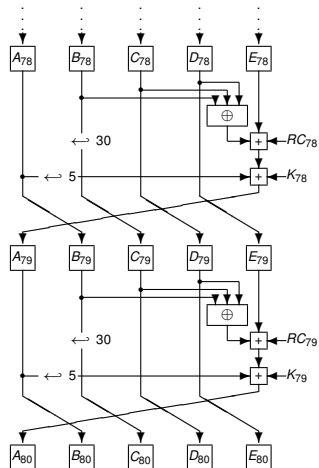
# Agenda

---

- Previous work: DFA on SHACAL1 (Ruilin Li et al., FDTC 2009)
- Extension to the SHA1 compression function
- DFA on SHA1COMPR
- Computational optimization
- Fault model
- Simulation results
- Conclusion and ongoing work

# SHACAL1

- SHACAL1 is a symmetric 160-bit block cipher
- $\text{SHACAL1} : \{0, 1\}^{160} \times \{0, 1\}^{512} \rightarrow \{0, 1\}^{160}$ ,  
 $(P, K) \mapsto C$
- It uses the building blocks of the SHA1 hash function without the final addition
- $P = A_0 || B_0 || C_0 || D_0 || E_0$
- $K = K_0 || K_1 || \dots || K_{15}$  and  
 $K_i = (K_{i-3} \oplus K_{i-8} \oplus K_{i-14} \oplus K_{i-16}) \ll 1$ ,  
 $i = 16, 17, \dots, 79$
- $C = A_{80} || B_{80} || C_{80} || D_{80} || E_{80}$



# DFA on SHACAL1 - Theoretical background

- SHACAL equation:  $(x \oplus \delta^{(j)}) - x = \Delta^{(j)}$ 
  - with up to  $m$  pairs  $(\delta^{(j)}, \Delta^{(j)})$  known,  $j = 1, \dots, m$
  - and  $x$  fixed and unknown.

⇒  $x$  can be determined by evaluating the carry flows of the addition:

⇒  $x_i = \delta_{i+1}^{(j)} \oplus \Delta_{i+1}^{(j)}$  if  $\delta_i^{(j)} = 1$ ,  $0 \leq i \leq 30$

⇒  $x$  can be determined (except for the highest bit) if  $\bigvee_{j=1}^m \delta^{(j)} = (*, 1, \dots, 1)$

# DFA on SHACAL1 (FDTC 2009)

- All shaded values are known to the attacker
- Induce errors in  $B_{77}$
- Determine  $D_{78}$  ( $= E_{79}$ ) using the SHACAL equation:

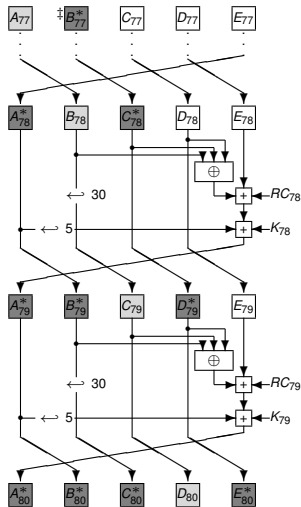
Taking the difference of the disturbed and undisturbed output of the  $\oplus$ -function yields:

$$x = BCD_{78} = B_{78} \oplus C_{78} \oplus D_{78}$$

$$\delta = C_{78}^* \oplus C_{78}$$

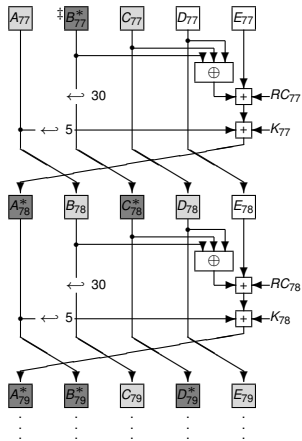
$$\Delta = (A_{79}^* - A_{79}) - ((A_{78}^* \leftrightarrow 5) - (A_{78} \leftrightarrow 5))$$

- Calculate  $K_{79}$  except for the highest bit
- Strip of the last round



# DFA on SHACAL1 (2)

- Determine  $D_{77}$  ( $= E_{78}$ ) using the SHACAL equation
- Calculate  $K_{78}$  (except for the highest bit)
- Repeat the steps above by inducing errors in  $B_{75}, \dots, B_{63}$
- Use the key derivation function to calculate all round keys
- Validate the correct one with one plain/cipher pair

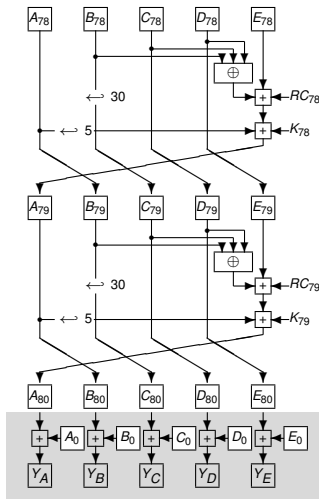


# SHA1 Compression function (SHA1COMPR)

- SHA1COMPR :
 
$$\{0, 1\}^{160} \times \{0, 1\}^{512} \longrightarrow \{0, 1\}^{160},$$

$$(P, K) \longmapsto Y$$
- $$\text{SHA1COMPR}(P, K) = Y_A || Y_B || \dots || Y_E :=$$

$$(A_{80} + A_0) || (B_{80} + B_0) || \dots || (E_{80} + E_0)$$
- Only interested in the case  $P = P(K_1)$  and  $K = K(K_2)$  with some unknown  $K_1$  and  $K_2$
- The state  $(A_{80} || B_{80} || C_{80} || D_{80} || E_{80})$  is not known to the attacker
- Use cases (fixed input, known output)
  - Key derivation functions
  - HMACs





# DFA on SHA1COMPR - Theoretical background

- SHA1 equation:

$$\begin{aligned} Y^{(j)} - X &= \Phi^{(j)} \\ (Y^{(j)} \leftarrow t) - (X \leftarrow t) &= \Psi^{(j)}, \end{aligned}$$

- given the rotation parameter  $t$
  - with  $\Phi^{(j)}$ ,  $\Psi^{(j)}$  known,  $Y^{(j)}$  random and unknown,  $j = 1, \dots, m$
  - and  $X = X_1 \cdot 2^{32-t} + X_0$  fixed and unknown.
- ⇒ By evaluation of the borrows at the borders you can determine  $X_1$  and  $X_0$  with falling probability from the left to the right

# DFA on SHA1COMPR

- Elimination of the final addition
- Induce errors on  $A_{79}$
- Build SHA1 equation:

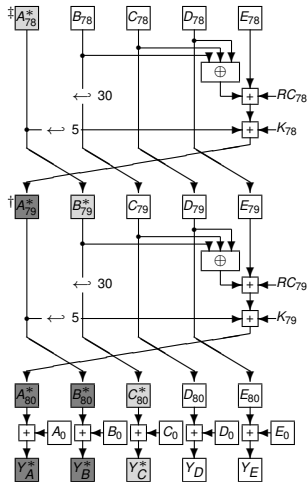
$$A_{79}^* - A_{79} = Y_B^* - Y_B$$

$$(A_{79}^* \leftrightarrow 5) - (A_{79} \leftrightarrow 5) = Y_A^* - Y_A$$

- Calculate remaining candidates for  $B_0$
- Insert errors on  $A_{78}$
- Build SHA1 equation to determine candidates for  $C_0$  independent of  $B_0$

$$(A_{78}^* \leftrightarrow 30) - (A_{78} \leftrightarrow 30) = Y_C^* - Y_C$$

$$(A_{78}^* \leftrightarrow 5) - (A_{78} \leftrightarrow 5) = Y_B^* - Y_B$$



## DFA on SHA1COMPR (2)

- Use SHACAL equation with

$$x = BCD_{79} = B_{79} \oplus C_{79} \oplus D_{79}$$

$$\delta = B_{79}^* \oplus B_{79} = ((Y_C^* - C_0) \leftrightarrow 2) \oplus ((Y_C - C_0) \leftrightarrow 2)$$

$$\Delta = (Y_A^* - Y_A) - (((Y_B^* - B_0) \leftrightarrow 5) - ((Y_B - B_0) \leftrightarrow 5)),$$

to drop all pairs  $(B_0, C_0)$  for which the equation has no solution and compute  $x = BCD_{79}$

- Notice that  $C_0$  will be completely determined in most cases  
But: Almost no impact on the number of candidates of  $B_0$
- Insert faults in  $A_{77}$
- Use SHA1 equation to determine candidates for  $D_0$
- Use SHACAL equation with  $x = BCD_{78}$  to reduce the candidates of  $D_0$

## DFA on SHA1COMPR (3)

- At the end of elimination phase we know:  $C_0$ ,  $D_0$ ,  $BCD_{79}$  und  $BCD_{78}$
- Calculate  $E_0$  and  $E_{79}$  (independent of  $B_0$ ):

$$\begin{aligned}E_0 &= Y_E - E_{80} = Y_E - D_{79} = \\ & Y_E - (BCD_{79} \oplus (((Y_C - C_0) \leftrightarrow 2) \oplus (Y_D - D_0))) \\ E_{79} &= BCD_{78} \oplus B_{78} \oplus C_{78} = \\ & BCD_{78} \oplus ((Y_D - D_0) \leftrightarrow 2) \oplus (Y_E - E_0)\end{aligned}$$

- Adapt attack of SHACAL1 to start in round 79 instead of round 80

- with  $E_{79}^* = E_{79} + Y_A^* - Y_A +$   
 $((Y_B - B_0) \leftrightarrow 5) - ((Y_B^* - B_0) \leftrightarrow 5) +$   
 $((Y_C - C_0) \leftrightarrow 2) \oplus (Y_D - D_0) \oplus (Y_E - E_0) -$   
 $((Y_C^* - C_0) \leftrightarrow 2) \oplus (Y_D^* - D_0) \oplus (Y_E^* - E_0)$

# Computational optimization (Impact of $B_0$ )

- $B_0$  influences

- $A_{79}$  and  $A_{79}^*$ , but not  $A_{79} - A_{79}^*$
- $K_{78}$ , but  $K_{78} + B_0 = \text{const}$  for all  $B_0$
- $E_{79}^* = E_{79} + Y_A^* - Y_A + ((Y_B - B_0) \leftrightarrow 5) - ((Y_B^* - B_0) \leftrightarrow 5) + \dots$

- Idea: If the 5 upmost bits of  $B_0$  are known,

- choose  $\hat{X} \in \{\text{min of all } B_0\text{'s, max of all } B_0\text{'s}\}$  and
- discard all errors  $Y_B^*$  equal to one of the remaining candidates of  $B_0$ , then

$$((Y_B - B_0) \leftrightarrow 5) - ((Y_B^* - B_0) \leftrightarrow 5) = ((Y_B - \hat{X}) \leftrightarrow 5) - ((Y_B^* - \hat{X}) \leftrightarrow 5)$$

⇒  $E_{79}^*$  does not "really" depend on  $B_0$

⇒ Only two values of  $B_0$  have to be tested

# Fault model

---

- Used a 32 bit faultmodel
- Also 16bit (and 8bit) fault model simulated
  - Needed less errors to determine  $B_0$ ,  $C_0$  and  $D_0$  with the SHA1 equation
  - Possibility to attack the final addition directly
  - But up to a factor of 2(4) more errors for the SHACAL equation
  - Costs less computation time (not critical)

# Simulation results (32 bit faultmodel)

Nr. of faults (elimination-phase)	remaining candidates (avg)					Nr. of faults (SHACAL1-phase)	success rate	computation time (avg)	Nr. of si- mulations
	$B_0$	$C_0$	$D_0$	$E_0$	$E_{79}$				
3 · 1622 = 4866	162406	1.8	1.7	7.7	22.3	9 · 15 = 135	93.8%	3.9 min	500
3 · 1288 = 3864	215834	1.6	1.9	6.8	25.4	9 · 15 = 135	93.0%	6.6 min	500
3 · 955 = 2865	291700	1.7	1.6	6.2	16.0	9 · 15 = 135	89.2%	7.5 min	500
3 · 622 = 1866	450078	1.8	1.7	6.3	18.0	9 · 15 = 135	88.0%	25 min	500
3 · 455 = 1365	565252	1.6	1.7	6.1	19.4	9 · 15 = 135	76.5%	82 min	153
3 · 289 = 867	954745	1.6	1.5	7.7	22.4	9 · 15 = 135	73.7%	171 min	114

- Impact of number of faults only to remaining candidates of  $B_0$ , the success rate and the computation time
- Memory overflows (to many candidates of  $B_0$ ) or timeout restrictions lowered success rate
- Change program to dynamic fault injection to get results independent of the success rate

# Conclusion and ongoing work

---

- With about 1000 faults it is possible to fully extract the secret inputs of the SHA1 compression function with high probability
- Work on SHA224/256 similar to the work done by Wei Yue-chuan et al. on SHACAL-2

(Wei Yue-chuan, Li Lin, Li Rui-lin and Li Chao, Differential Fault Analysis on SHACAL-2, Journal of Electronics and Information Technology, 2010)



# The End

---

- Thank you for your attention

