Side-Channel Attacks on SHA-1-based Product Authentication ICs

David Oswald*

The University of Birmingham, UK
d.f.oswald@cs.bham.ac.uk

Abstract. To prevent product counterfeiting, a common practice is to cryptographically authenticate system components (e.g., inkjet cartridges, batteries, or spare parts) using dedicated ICs. In this paper, we analyse the security of two wide-spread examples for such devices, the DS28E01 and DS2432 SHA-1-based authentication ICs manufactured by Maxim Integrated. We show that the 64-bit secret can be fully extracted using non-invasive side-channel analysis with 1,800 and 1,200 traces, respectively. Doing so, we present the, to our knowledge, first gray-box side-channel attack on real-world devices employing an HMAC-like construction. Our results highlight that there is an evident need for protection against implementation attacks also for the case of low-cost devices like product authentication ICs

Keywords: Side-channel analysis, SHA-1, product authentication, anti-counterfeiting, real-world attack

1 Introduction

Counterfeit electronic products have become an immense problem for manufacturers. According to a report of the United Nations Office on Drugs and Crime [20], the market for counterfeit goods had a value of USD 250 billion in 2012. Approximately 8% of all counterfeit products are electrical or computer equipment (based on the number of counterfeit seizures made at the European borders in 2008). Hence, protecting products against being "cloned" is a necessity for a manufacturer today.

Devices that consist of several components of different complexities appear to be a profitable target for counterfeit in particular: For example, while fake printers are relatively rare, there is a huge variety of compatible ink cartridges for all brands, presumably because cartridges are easy to produce and in constant demand. The same holds for similar low-cost items like accessories for mobile phones (e.g., chargers, batteries, etc.) and also for more expensive equipment like medical sensors or extension modules for network infrastructure. To ensure

* Part of this work was carried out while the author was at the Chair for Embedded Security, Prof. Dr.-Ing. Christof Paar, Ruhr-University Bochum, Germany.

that such components are genuine, various commercial solutions based on cryptographic authentication are available. Usually, an additional IC is placed on the device to be authenticated. The host (e.g., a printer or a mobile phone) then executes an authentication protocol with the IC to verify that the component (e.g., an inkjet cartridge or a battery) is genuine. The cryptographic algorithms commonly encountered in this area range from AES [1] and SHA-1 [12] over SHA-2 to Elliptic Curve Cryptography (ECC) [9]. Commonly, the authentication is unilateral, i.e., the device is authenticated to the host, but not vice versa.

Being often relatively low-cost products, devices protected with such ICs are easily available to a potential adversary for detailed analysis. Hence, the question about the physical security arises. In this paper, we focus on the SHA-1 EEPROM product line of Maxim Integrated, analysing two specific ICs from a side-channel point-of-view, the DS28E01-100 [12] and the older DS2432 (which is not recommended for new designs). These devices enable the unilateral authentication of a component to the host using a shared 64-bit secret in a challenge-response protocol based on SHA-1 as the main cryptographic primitive. Note that more expensive invasive and semi-invasive attacks (e.g., microprobing, circuit modification with a Focused Ion Beam (FIB), laser Fault Injection (FI), etc.) are outside the scope of this paper. Instead, we focus on Side-Channel Analysis (SCA) that can be performed using relatively low-cost oscilloscopes (in the range of a few thousand EUR) or even cheaper, specialised acquisition hardware.

1.1 Related Work

In contrast to standard block ciphers, implementation attacks on SHA-family hash functions have to our knowledge so far mostly been studied theoretically or for prototypical implementations: In [10], an FI attack on the SHA-1-based cipher SHACAL-1 is proposed, which is extended to also apply for a standard SHA-1 Hash-based Message Authentication Code (HMAC) in [8]. With respect to SCA, McEvoy et al. described a Correlation Power Analysis (CPA) on their own implementation of a SHA-2 HMAC on an FPGA [13], also covering suitable countermeasures against this type of attack. In [7], template attacks on HMACs are studied. The authors of [2] generalize and improve the ideas of [13].

With regard to real-world targets, in a presentation at the 27th Chaos Communication Congress [4], a sophisticated FI-based attack on an older SHA-1 device, the Dallas iButton, was described. The author also disclosed information on the authentication protocol, which is similar to that of our Devices Under Test (DUTs). The attacks of [4] are based on partially overwriting the secret (achieved using FI) and may also apply to the DS2432 or the DS28E01 analyzed in this paper. However, the attacks could be rather easily prevented by setting the corresponding write-protect flag for the memory storing the secret.

1.2 Contribution

As the main contribution, we present the—to our knowledge—first real-world SCA of SHA-1-based authentication ICs. In doing so, we devise a method to

further reduce the attack complexity (in terms of the number of targeted rounds), using properties of the way the SHA-1 is employed in the given context. This paper is partially based on the research done for the author's PhD thesis [15].

The remainder of this paper is structured as follows: In Section 2, we describe the authentication protocol used by the DS28E01 and DS2432. Section 3 outlines a basic attack, which is subsequently extended to exploit certain properties of the W-schedule of SHA-1. The described methods are then applied in practice to the DUTs in Section 4. Finally, we conclude in Section 5, covering future work, responsible disclosure, and potential countermeasures.

2 Authentication Protocol

As an initial step, based on the full datasheet for the older DS1961S iButton (which has a similar protocol) and source code for the communication with the DS2432 found on the Internet, we understood and implemented the full communication protocol with the DS28E01 and DS2432. It turned out that the protocol for the DS28E01 only differs in details from that of the DS2432.

On the electrical level, the DUTs use Maxim's 1-wire interface [11]. A single supply/data pin (`IO`) is at the same time used for delivering the operating voltage and bidirectional communication. This is achieved by connecting the supply voltage via a small pull-up resistor (in the range of $1\,\mathrm{k}\Omega$) to `IO` and actively pulling the line low for data communication in an open-drain configuration. In addition, the interface requires a ground connection, hence, technically, two wires are used.

The 1-wire interface allows bit rates of $15.3\,\mathrm{kBit/s}$ ("regular speed") and $125\,\mathrm{kBit/s}$ ("overdrive speed"), respectively. For each bit, the interface uses a separate time slot. The duration for which the data line is pulled low determines whether a one or zero is sent. For reading data, the host issues a "read" slot and then disconnects its driver to bring the data line its default (high-impedance) state. To send a zero, the DUT then pulls `IO` low for a certain duration, otherwise, for a one, `IO` is left at a high level.

Both the DS28E01 and the DS2432 employ a straightforward challenge-response protocol to prove the authenticity of the device. To this end, the host writes a 5-byte (for the DS28E01) or 3-byte (for the DS2432) challenge to an internal buffer ("scratchpad memory") of the device and sends a `ReadAuthPage` command. The DUT then computes a slightly modified SHA-1 hash over the data stored in the addressed memory page, the Unique Identifier (UID) of the DUT, the challenge, certain constants, and the 64-bit secret k. The result is returned to the host as response. For both DUTs, the function SHA-1' follows the standard [14], except for the fact that only one block is hashed and the addition of the final $H_0^{(i-1)}$, ... is omitted (cf. [14, p. 19, step 4]). The overall protocol is shown in Figure 1.

The function f is essentially a simple concatenation: the input to the SHA-1 is constructed as shown in Table 1, whereas k_i $(0 \leq i \leq 7)$ are the eight bytes of the secret, P_i $(0 \leq i \leq 31)$ the bytes of the addressed page of the DUT's

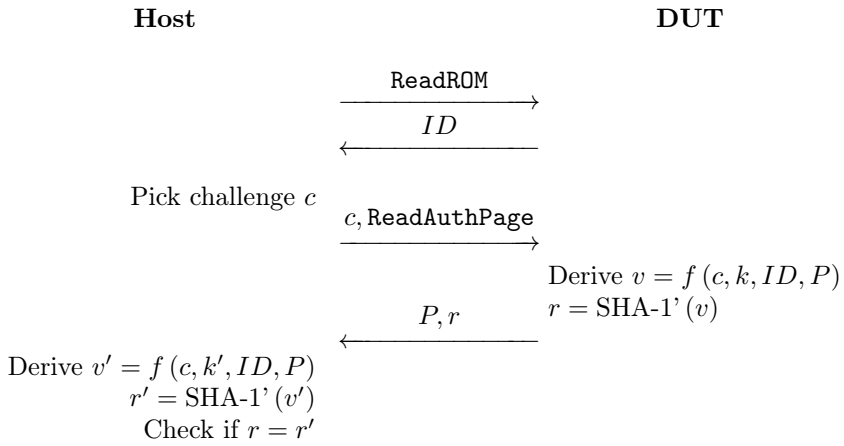

Fig. 1: Simplified protocol for authenticating DS28E01/DS2432 to host

memory, M a value derived from the page address, ID_i $(0 \leq i \leq 6)$ the UID, c_i $(0 \leq i \leq 4)$ the 5-byte challenge, and x_i $(0 \leq i \leq 10)$ fixed constants. For the DS2432, the first two challenge bytes c_0 and c_1 are set to `0xFF`, because the challenge has a length of only three byte for this device.

Word	Byte 3	Byte 2	Byte 1	Byte 0
W_0	k_0	k_1	k_2	k_3
W_1	P_0	P_1	P_2	P_3
W_2	P_4	P_5	P_6	P_7
		...		
W_8	P_{28}	P_{29}	P_{30}	P_{31}
W_9	c_0	c_1	x_0	x_1
W_{10}	M	ID_0	ID_1	ID_2
W_{11}	ID_3	ID_4	ID_5	ID_6
W_{12}	k_4	k_5	k_6	k_7
W_{13}	c_2	c_3	c_4	x_2
W_{14}	x_3	x_4	x_5	x_6
W_{15}	x_7	x_8	x_9	x_{10}

Table 1: Input to the SHA-1 for the `ReadAuthPage` command of the DS28E01 (similar for DS2432)

Incidentally, the length of 3 byte is short enough that an adversary could get the full dictionary of challenge-responses pairs (for one particular device) in reasonable time: for setting up a challenge and receiving the response, 296 bit are exchanged between host and device, which leads to approximately 19.2 ms in normal mode (65 µs per bit) and 2.4 ms in overdrive mode (8 µs per bit) in the

ideal case. Additional delays for start-up and the SHA-1 execution add another 3 ms, resulting in an overall best-case figure of 5.4 ms per challenge-response pair. Hence, to obtain all 2^{24} pairs, approximately 1 day of communication with the DUT and 368 MB of storage would be required. In contrast, obtaining the full dictionary for the DS28E01 would take approximately 188 years under the above conditions.

Note that we did not thoroughly analyse the mathematical security of the employed protocol. However, for the given application where only one block is hashed and hence length-extension attacks [16] do not apply, using the SHA-1 without a proper HMAC construction [3] (which would require two SHA-1 executions) seems to be "secure enough".

3 Side-Channel Analysis of SHA-1

In contrast to block ciphers like AES or DES, SHA-1 involves mostly linear operations and does not have separate, constant subkeys combined with varying input. Instead, for the present DUT, the secret key is part of the input. Hence, in order to apply SCA to extract the secret in the given situation, a dedicated attack procedure had to be devised.

3.1 Basic Approach

Based on the SCA on a SHA-1 HMAC proposed in [13], we first started with a basic attack (which is a similar to the method independently proposed in [2] for SHA-2). This method was subsequently extended to reduce the number of targeted rounds and hence lower the computational complexity and reduce the susceptibility to errors. In the present case, the output of the SHA-1 (after 80 rounds) is available to the adversary, while only part of the input (the challenge) can be chosen. Hence, our attack first targets and undoes the final round and is then repeated for prior rounds until enough information to recover the secret is available.

In the following, we denote the value of the SHA-1 32-bit state registers after round i as A_i, B_i, C_i, D_i, and E_i, respectively. Hence, the output of the SHA-1 available to the adversary is $(A_{79}, B_{79}, C_{79}, D_{79}, E_{79})$. From this, the four state words A–D after round 78 can be directly computed as $A_{78} = B_{79}$, $B_{78} = \mathrm{rrot}_{30}(C_{79})$, $C_{78} = D_{79}$, and $D_{78} = E_{79}$. In contrast, to compute the remaining register E_{78}, the knowledge of the (unknown and secret-dependent) value W_{79} is required:

$$E_{78} = A_{79} - K_{79} - W_{79} - \mathrm{lrot}_5(B_{79}) - F_{79}(C_{79}, D_{79}, E_{79})$$

Note that W_{79} depends on the challenge and hence cannot be directly recovered using CPA. However, by construction of the W schedule of SHA-1, W_{79} can be written as a XOR combination of a known (and challenge-dependent) value W_{79}^{known} and an unknown value W_{79}^{secret} depending on the secret. Thus, considering all candidates for W_{79}^{known} and identifying the correct value with CPA, E_{78}

can be fully recovered and the complete round be inverted. Since W_{79}^{known} is a 32-bit value, 2^{32} candidates would have to be tested with SCA, which is possible but could be undesirable in certain cases. However, using partial correlations for 8-bit parts (starting at the least-significant byte) as suggested in [13], the number of candidates can be reduced at the cost of a higher trace complexity. Having recovered W_{79} and inverted the final round, the attack now identically repeats for round 78, 77, and so on. Following [8, 2], to fully undo the W schedule and recover the complete input including the secret, the sixteen values $W_{79}, \ldots, W_{64}$ are sufficient. In total, this attack hence requires $16 \cdot 4$ CPAs with 2^8 candidates each, or alternatively 16 CPAs with 2^{32} candidates each.

3.2 Improved Attack on Final Two Rounds

While the basic method is fully practical for the present DUT, it has the shortcoming that a single error in one round will affect all subsequent rounds and make the attack fail. Since we target relatively linear operations (addition modulo 2^{32}), the occurrence of such errors is more likely compared to non-linear S-boxes, especially for earlier rounds where the leakage is partially lower than for the final rounds. However, in the following we show that (for the way the input is constructed for the DUT) the knowledge of W_{79}^{secret} and W_{78}^{secret} is sufficient, i.e., only the final 2 rounds have to be targeted.

First, note that when fully unrolling the W schedule, W_{78}^{secret} and W_{79}^{secret} are linear combinations of several rotated instances of W_0 and W_{12}. The question arises if W_0 and W_{12} can be uniquely recovered from W_{78}^{secret} and W_{79}^{secret}. To this end, we express left-rotation by j positions as polynomial multiplication with x^j modulo $x^{32}+1$ [17] and denote the polynomial representation of a word in $\mathbb{F}_2[X]/(x^{32}+1)$ with the same letter in lower case. Then,

$$\begin{aligned} w_{79}^{\text{secret}} &= \left(x^{22}+x^8\right) \cdot w_0 + \left(x^{18}+x^{14}+x^{12}+x^8+x^6\right) \cdot w_{12} \\ &= a_1 \cdot w_0 + b_1 \cdot w_{12} \\ w_{78}^{\text{secret}} &= \left(x^{20}+x^{18}+x^{15}+x^8+x^7\right) \cdot w_0 + \left(x^{15}+x^{11}+x^8\right) \cdot w_{12} \\ &= a_2 \cdot w_0 + b_2 \cdot w_{12} \end{aligned}$$

This linear equation system is solvable if the inverse d^{-1} of the determinant

$$d = det \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}$$

exists. This is the case for the given scenario, with $d^{-1} = x^{29}+x^{26}+x^{24}+x^{23}+x^{21}+x^{20}+x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^8+x^4+x^2+x$. Then, we obtain

$$\begin{aligned} w_0 &= \left(w_{79}^{\text{secret}} \cdot b_2 + w_{78}^{\text{secret}} \cdot b_1\right) \cdot d^{-1} \\ w_{12} &= \left(w_{79}^{\text{secret}} \cdot a_2 + w_{78}^{\text{secret}} \cdot a_1\right) \cdot d^{-1} \end{aligned}$$

Note that this method also applies when only one round is inverted by means of SCA. Then, one exhaustively tests all 2^{32} candidates for W_{78}^{secret}, applying the

above method to obtain w_{12} and w_0, and checking the resulting secret with one SHA-1 output.

4 Practical Results

For evaluating the practical applicability of the above attack to the DUTs, we implemented the 1-wire protocol on a custom device (based on an FPGA) for precise control over the protocol execution. We then built simple test fixtures to access the pins of the DUT and insert a measurement resistor into the ground line (490 Ω for the DS28E01, 50 Ω for the DS2432). Note that the DS28E01 continued to function correctly even though a relatively high resistor value was chosen. We used a Picoscope 6402C to record the voltage drop over the measurement resistor at a sample rate of 625 MSPS. We digitally downsampled the resulting traces by a factor of 5, leading to a sample rate of 125 MSPS, and furthermore lowpass-filtered the traces with a cutoff frequency of 8 MHz (DS28E01) and 5 MHz (DS2432), respectively. These parameters were determined heuristically by observing the correlation for a known secret in the profiling phase (Section 4.1). We found that the SHA-1 is executed after the 32-byte page data, one constant byte `ff`, and the (inverted) 2-byte Cyclic Redundancy Check (CRC) has been read. Hence, we triggered the trace acquisition on the final bit of the CRC being read.

4.1 Profiling

Using the described setup, we recorded 3,000 traces each for a fixed, known secret, using uniformly distributed, random challenges. We then performed CPAs for various intermediate values and leakage models. Experimentally, we found that the leakage of both DUT follows the Hamming Distance (HD) between the SHA-1 state registers in subsequent rounds, suggesting a complete hardware implementation. Figure 2 and Figure 3 depict the correlation for the DS28E01

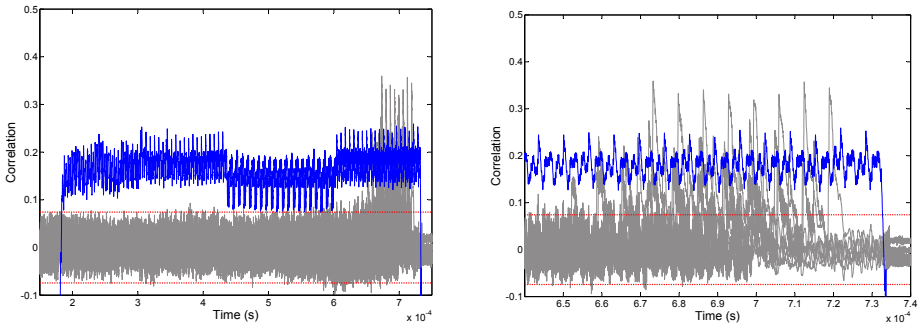

Fig. 2: DS28E01: correlation for HD between states of E in final 8 rounds after 3,000 traces (grey); average trace (blue, not to scale). Left: overview, right: zoomed on final rounds

and the DS2432 for the pairwise HD between the final eight states of E, i.e., HD (E_{78}, E_{79}), HD (E_{77}, E_{78}), and so on. The average trace (amplitude not to scale) is overlaid in blue. The red horizontal lines indicate the expected noise interval of $\pm 4/\sqrt{\#\,\text{traces}}$. For the DS28E01, the correlation for the final two rounds reaches approximately 0.35, while for the DS2432, a value of approximately 0.49 is observed. This is likely due to the DS2432 using an older process technology with higher current consumption, leading to a higher overall Signal to Noise Ratio (SNR)—even with a much lower measurement resistor value.

In the average traces, the 80-round structure of the SHA-1 is clearly visible, and even the different boolean functions are discernible. For instance, at approximately 450 μs (for the DS28E01), the shape of the trace changes significantly. This region comprises the execution of rounds 40 to 59, in which $F_i = (B \,\&\, C) \mid (B \,\&\, D) \mid (C \,\&\, D)$.

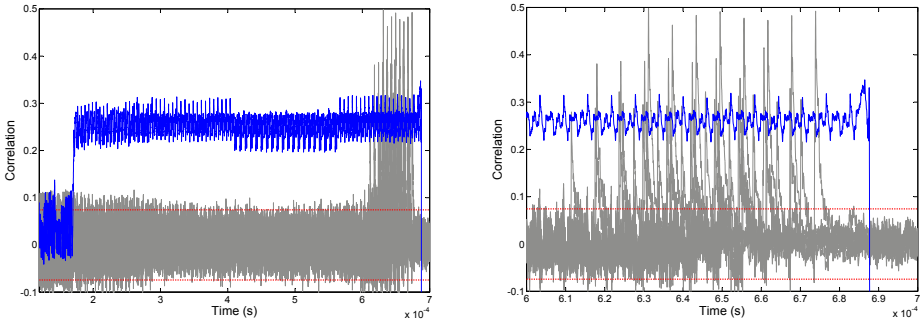

Fig. 3: DS2432: correlation for HD between states of E in final 8 rounds after 3,000 traces (grey); average trace (blue, not to scale). Left: overview, right: zoomed on final rounds

4.2 Full Key Recovery

Before applying the attack procedure of Section 3 to the traces acquired for the DUTs, we estimate the expected correlations based on the profiling results (Section 4.1). When recovering the Least Significant Byte (LSByte), 8 out of 32 bit are predicted, hence, we expect a correlation of $0.35 \cdot \sqrt{8/32} = 0.175$ (DS28E01) and 0.245 (DS2432), respectively [5].

Carrying out the actual key recovery, we obtained correlations that closely match the expected values, for instance, values between 0.175 and 0.177 for the LSByte in case of the DS28E01 and 0.268 to 0.279 for the DS2432. We then successfully recovered the full key for both DUTs attacking the final two rounds.

To more precisely estimate the amount of required traces, we computed the average Partial Success Rate (PSR) for single bytes and the Global Success Rate (GSR) [19] for the recovery of the full secret for 16 sets of 3,000 traces each for the DS28E01 (with 16 different random secrets).

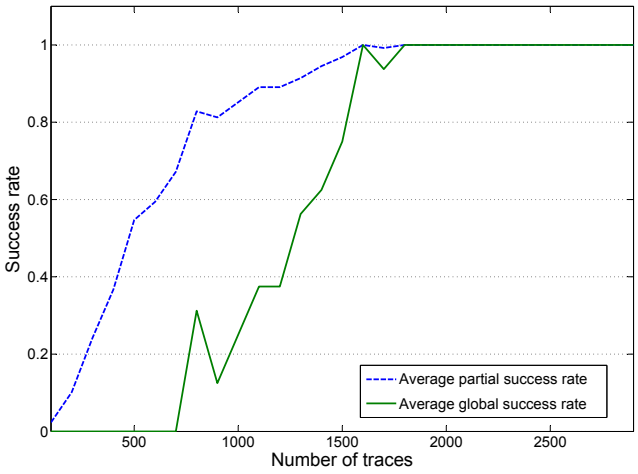

Fig. 4: Average GSR and average PSR for DS28E01, 16 experiments

The results are depicted in Figure 4. The average PSR is computed for each byte in round 78 and 79 separately. Note that in contrast to block ciphers, however, a failure in correctly recovering a single byte will cause all subsequent bytes to fail as well. Hence, the GSR is a more appropriate metric in this case, as it only "accepts" secrets that were recovered completely. A stable GSR of 1 is reached after 1800 traces, with only a single experiment failing at 1700 traces. The acquisition of 1800 traces took approximately 40 min. with our setup. However, note that the rate was mainly limited due to the oscilloscope and PC storage (not the operation of the DUT) and could be significantly optimized further (e.g., by choosing a lower sample rate, acquiring only the relevant part of the traces, and so on). We also computed the Partial Guessing Entropy (PGE) (over

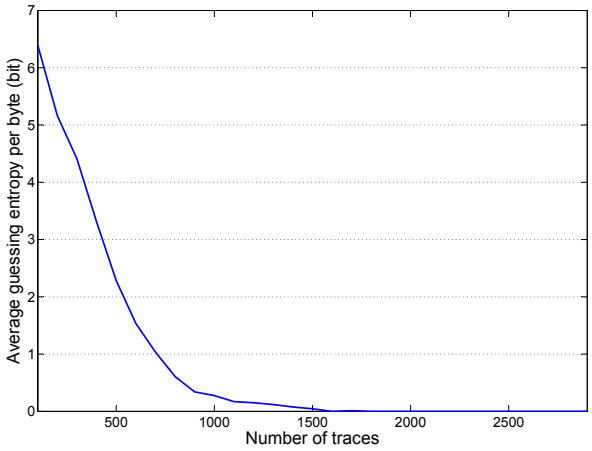

Fig. 5: PGE in bit per byte for DS28E01, 16 experiments

both rounds for 4 byte each) as shown in Figure 5. The PGE falls below 1 bit (i.e., the correct candidate has an expected rank of 1 or 2) after approximately 700 traces. This would mean that after recovering the LSByte with a CPA with 2^8 candidates, 2 candidates remain on average. These two candidates can then be checked with two 2^8-candidate CPAs for the next byte and so on, leading to in total $2^8 + 2 \cdot 2^8 + 2^2 \cdot 2^8 + 2^3 \cdot 2^8 = 15 \cdot 2^8 = 3840$ candidates to be checked with CPA per round.

Note that due to the similarity to the DS28E01, we did not compute the success rate over many experiments for the DS2432. However, due to the larger leakage, the key recovery can be expected to require even less traces in this case. We verified this assumption by performing the attack for one fixed key on the DS2432, and found that the correct candidate reached rank 1 after 500 traces for 7 of 8 byte—only for byte 2 in the final round, the correct candidate moved to rank 2 after 1,100 traces, requiring at least 1,200 traces to be stable at rank 1. Based on this experiment and the higher correlation obtained for the DS2432, we hence assume a value of 1,200 traces to be a good upper-bound estimate for the security of this DUT.

5 Conclusion and Outlook

In this paper, we presented successful key recovery attacks on Maxim's DS28E01 and DS2432 product authentication ICs. The methods allow to fully extract the 64-bit secret with approximately 40 min. for the trace acquisition. The employed SCA techniques have relatively low requirements with respect to the measurement equipment (sample rate 125 MSPS) and the trace complexity (1,800 and 1,200 traces, respectively). Hence, it is conceivable that the attacks could also be carried out using low-cost tools, e.g., the GIAnT or the ChipWhisperer, both featuring ≥ 100 MSPS Analog to Digital Converters (ADCs) [18, 6].

Hence, SCA may pose a serious problem even given that the DUTs are a low-cost solution to provide some protection against counterfeiting and are of course not intended to replace high-security smartcard ICs. We had a brief look at newer SHA-2-based authentication ICs from two manufacturers, including Maxim Integrated. Figure 6 depicts the respective side-channel traces: for DUT 1 (left), we measured the Electro-Magnetic (EM) emanation (due to certain properties of the available test board), while we acquired normal current measurements for DUT 2.

Although we did not perform a thorough analysis as demonstrated for the DS28E01/DS2432, applying SCA to these newer DUTs appears to be complicated by a higher amount of noise both in the signal amplitude and timing. However, more precisely analyzing SHA-2-based devices with regard to their level of protection against SCA is an interesting point for future work.

Apart from that, the general question on how to appropriately protect against the demonstrated attacks arises. For existing designs using the analysed ICs, certain steps on the system level to mitigate the consequences of a successful key recovery should be taken: First of all, it should be ensured that there are

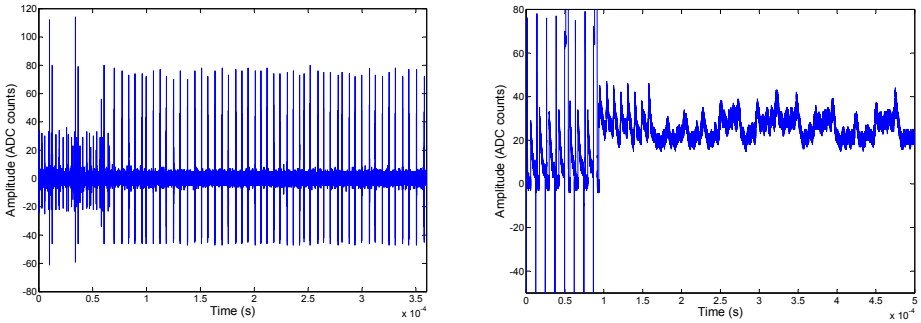

Fig. 6: Part of example traces for SHA-2 ICs from different manufacturers, left: EM, DUT 1, right: power, DUT 2

no system-wide secrets (stored on every ICs) and that secure key diversification (e.g., based on the UID of the DUT) is in place. Otherwise, a single successful attack on a single device would render all other devices insecure. In this regard, note that also the counterpart on the host must be protected, especially if it stores a system-wide diversification key.

Checking the UID (and using it for key diversification) on the host also ensures that an adversary cannot use a real DS28E01/DS2432 IC for a cloned product by simply copying the recovered secret and memory contents. Instead, since the UID is factory-programmed, he would have to create a custom emulator, e.g., using a microcontroller or a custom ASIC. This increases the complexity and cost of counterfeiting, possibly to a point where the cloning becomes unprofitable.

In the long run, ICs like the DUTs analysed in this paper should include side-channel countermeasures to at least prevent low-cost SCA techniques and raise the bar in terms of trace and measurement equipment complexity. Adapting common countermeasures (e.g., randomization of timing, masking, etc.) to the specific requirements for product authentication ICs (in particular low cost and hence low chip area) is an interesting problem.

Finally, having discovered the security problems, as part of a responsible disclosure process, we contacted the vendor Maxim and informed them about our investigations. Maxim acknowledged our result and is exploring ways to mitigate the security issues. We would also like to note that the more recent products of Maxim are not directly vulnerable to the methods presented in this paper.

References

1. Atmel. ATAES132A 32K AES Serial EEPROM Specification. Datasheet, July 2015. `http://www.atmel.com/Images/Atmel-8914-CryptoAuth-ATAES132A-Datasheet.pdf`.
2. S. Belaid, L. Bettale, E. Dottax, L. Genelle, and F. Rondepierre. Differential Power Analysis of HMAC SHA-2 in the Hamming Weight Model. In *SECRYPT'13*, Reykjavik, Iceland, July 2013. Scitepress.

3. M. Bellare, R. Canetti, and H. Krawczyk. Keying Hash Functions for Message Authentication. In *Advances in Cryptology – CRYPT0'96*, LNCS, pages 1–15, London, UK, UK, 1996. Springer.
4. C. Brandt. Hacking iButtons. Presentation at 27C3, 2010. `http://cribert.freeforge.net/27c3/ibsec.pdf`.
5. E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES'04*, volume 3156 of *LNCS*, pages 16–29. Springer, 2004.
6. Colin O'Flynn. ChipWhisperer. Website, July 2015. `https://www.assembla.com/spaces/chipwhisperer/wiki`.
7. P.-A. Fouque, G. Leurent, D. RÃľal, and F. Valette. Practical Electromagnetic Template Attack on HMAC. In C. Clavier and K. Gaj, editors, *CHES'09*, volume 5747 of *LNCS*, pages 66–80. Springer, 2009.
8. L. Hemme and L. Hoffmann. Differential Fault Analysis on the SHA1 Compression Function. In *Proceedings of the 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography – FDTC'11*, LNCS, pages 54–62, Washington, DC, USA, 2011. IEEE Computer Society.
9. Infineon. ORIGA SLE95200. Datasheet, July 2015. `http://www.infineon.com/dgdl/ORIGA2_SLE95200_Product_Brief_v1+00.pdf?fileId=db3a30433580b3710135a50170336cd8`.
10. R. Li, C. Li, and C. Gong. Differential Fault Analysis on SHACAL-1. In *Proceedings of the 2009 Workshop on Fault Diagnosis and Tolerance in Cryptography – FDTC'09*, pages 120–126, Washington, DC, USA, 2009. IEEE Computer Society.
11. Maxim Integrated. 1-Wire. Website, July 2015. `http://www.maximintegrated.com/en/products/comms/one-wire.html`.
12. Maxim Integrated. DS28E01-100 1Kb Protected 1-Wire EEPROM with SHA-1 Engine. Website, July 2015. `http://www.maximintegrated.com/en/products/digital/memory-products/DS28E01-100.html/tb_tab0`.
13. R. McEvoy, M. Tunstall, C. C. Murphy, and W. P. Marnane. Differential power analysis of HMAC based on SHA-2, and countermeasures. In *Proceedings of the 8th international conference on Information security applications*, WISA'07, pages 317–332. Springer, 2007.
14. NIST. FIPS 180-4 Secure Hash Standard (SHS). `http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf`.
15. D. Oswald. *Implementation Attacks: From Theory to Practice*. PhD thesis, Ruhr-University Bochum, September 2013.
16. B. Preneel and P. C. Oorschot. MDx-MAC and Building Fast MACs from Hash Functions. In D. Coppersmith, editor, *Advances in Cryptology – CRYPT0'95*, volume 963 of *LNCS*, pages 1–14. Springer, 1995.
17. R. L. Rivest. The Invertibility of the XOR of Rotations of a Binary Word. *Int. J. Comput. Math.*, 88(2):281–284, Jan. 2011.
18. Sourceforge. GIAnT (Generic Implementation ANalysis Toolkit). Website, April 2013. `https://sf.net/projects/giant/`.
19. F.-X. Standaert, T. G. Malkin, and M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In A. Joux, editor, *EUROCRYPT'09*, volume 5479 of *LNCS*, pages 443–461. Springer, 2009.
20. United Nations Office on Drugs and Crime. Counterfeit Goods - A bargain or a costly mistake? Fact Sheet, 2013. `http://www.unodc.org/documents/toc/factsheets/TOC12_fs_counterfeit_EN_HIRES.pdf`.