Side-Channel Analysis of Keymill

Christoph Dobraunig, Maria Eichlseder, Thomas Korak, and Florian Mendel

IAIK, Graz University of Technology, Austria

Abstract. Keymill is a side-channel resilient key generator, also known as re-keying function. Re-keying functions are a crucial building block of fresh re-keying schemes. To ensure the security against side-channel analysis of re-keying schemes, the used re-keying function has to withstand both simple power analysis and differential power analysis. We present a DPA attack on Keymill, which relies on the assumption that the dynamic power consumption of a digital circuit is tied to the 0 → 1 and 1 → 0 switches of its logical gates. Hence, the power consumption of the shift-registers used in Keymill depend on the 0 → 1 and 1 → 0 switches of its internal state. This information is enough to obtain the internal differential pattern (up to a small number of bits, which have to be brute-forced) of the 4 registers of Keymill after the nonce (or IV) is absorbed.

Keywords: side-channel analysis · re-keying

1 Introduction

Fresh re-keying [5] is an approach for precluding differential power analysis (DPA) on cryptographic primitives. Fresh re-keying follows a separation-of-duties principle, where a re-keying function takes the burden of protecting against DPA away from the re-keyed cryptographic primitive by processing a nonce and master key to always compute a fresh session key, while the latter fulfills its original cryptographic mission. Therefore, the used re-keying function has to provide resistance against SPA and DPA attacks, either by its design, or by application of countermeasures like threshold implementations [6], masking [7], hiding [2], re-shuffling [4], etc.

The re-keying function Keymill [8] claims SCA-security by design. Keymill consists of 4 shift-registers, where each acts as input for a non-linear function (taken from Achterbahn [3]). The outputs of those non-linear functions act as inputs for the shift-registers. However, the outputs are connected via a rotating cross-connect to the inputs of the 4 shift-registers. This cross-connect joins function outputs with shift-register inputs cyclically per clock. For this construction and also for a toy example consisting of two 8-bit registers involving a similar rotating cross-connect, the authors claim that no DPA-attack is possible without making a hypothesis for the whole key, or equivalently for the whole internal state of the registers.

In this work, we show that this approach is not sufficient, and present a DPA attack against Keymill. The attack we propose works without making a hypothesis about concrete values of the state, or secret key. Instead, we recover the internal difference of neighboring bits of the shift-registers. As observed in [1,9], the dynamic power consumption of shift-registers depends on the number of internal differences of neighboring bits. The more internal differences we have, the more power the shift-register consumes. We recover those internal differences by comparing the power consumption of a reference nonce with power traces of a modified nonce, where a single difference has been injected.

2 Brief Description of Keymill

Keymill [8] is new keystream generator recently proposed by Taha, Reyhani-Masoleh and Schaumont at SAC 2016. It operates on an internal state of 128 bits, composed of 4 NLFSRs as shown in Fig. 1. Register R_0 is 31 bits, registers R_1 and R_2 are 32 bits, and register R_3 has 33 bits. The feedback functions F_0, F_1, F_2 and F_3 are selected from those proposed for the stream cipher Achterbahn [3]. The feedback functions are mixed via a rotating cross-connect, depending on the current clock cycle index i:

$$F_k \to R_{k+i \pmod 4} \quad \text{for } k = 0, 1, 2, 3.$$

After loading the 128-bit secret key into the internal state, 4 bits of the 128-bit IV that can be monitored (or controlled) by the attacker are added to the feedback functions of the registers in each clock cycle. After absorbing the IV in 16 clock cycles the internal state is clocked 33 more times before producing any output. Afterwards 4 bits of output are generated (one from each register) in each clock cycle. We refer to the specification [8] for a more detailed description of Keymill.

3 Side-Channel Attack on Keymill

In this section, we will show attacks on Keymill. First of all, we give some insight in the power consumption of shift-registers and show how this power consumption can be used to recover differences of neighboring shift-register bits. This technique, together with the fact that the first bits of the shift-registers are not used in the feedback function of Keymill, allows us to mount a side-channel attack. For simplicity, we first demonstrate the attack on a modification of Toy Model II given in the Keymill specification [8], and afterwards discuss the necessary adaptations for attacking Keymill.

3.1 Power Consumption of a Shift-Register

In the following attacks, we exploit the dynamic power consumption of the shift-registers at the triggering edge of the clock (i.e., positive edge). More specifically, we observe the dynamic power consumption of the building blocks of the shift-registers, the D-flip-flops. As elaborated by Zadeh and Heys [9], the dynamic

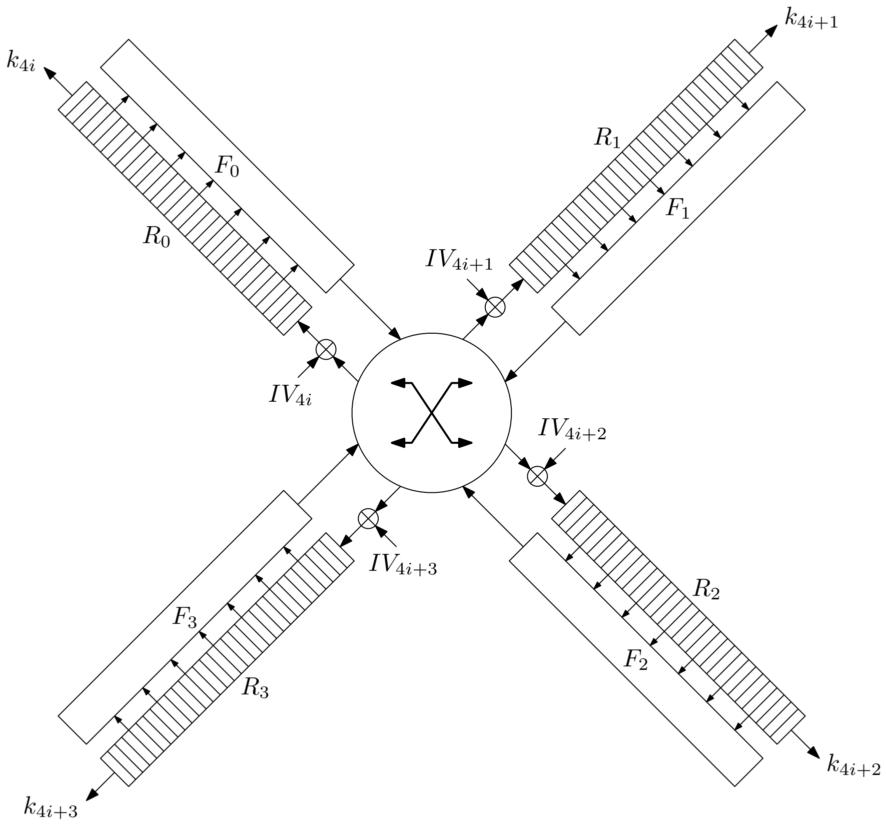

Fig. 1: Structure of Keymill

power consumption of a D-flip-flop at the triggering edge depends on whether its state changes. If the state of the D-flip-flop changes, more power is consumed than if it remains unchanged. As an example, Zadeh and Heys [9] analyze a D-flip-flop constructed out of 6 NAND gates. For such a flip-flop, 3 gates change if the flip-flop changes its state, whereas only one gate changes if not.

Next, we have a look at the power consumption of a shift register. For simplicity, consider a 4-bit shift register consisting of 4 flip-flops D_0, D_1, D_2, and D_3. In the following, we assume that D_4 is the input of our registers, which is shifted towards D_0. For instance, let us consider the power consumption of the change from state $S_0 = \texttt{0110}_2$ to state $S_1 = \texttt{1101}_2$. For this transition, D_0 changes its state, D_1 keeps its state, D_2 changes its state, and D_3 changes its state. Since the power consumption of the flip-flops is higher if they change their state, the power consumption of the shift register is correlated with the Hamming weight of $S_0 \oplus S_1$. In this example, 3 flip-flops change their state.

Now, we want to consider a state change from S_0 to S_1', where we shift in a `0` instead of a `1` as before. So we observe the power consumption for the change from state $S_0 = \texttt{0110}_2$ to state $S'_1 = \texttt{1100}_2$. If this transition happens, only two flip-flops change their state. Thus, we observe for the transition $S_0 \rightarrow S'_1$ a smaller power consumption than for $S_0 \rightarrow S_1$. This allows us to derive information about the difference of the bits stored in D_4 and D_3 of S'_1 and S_1, respectively. In more detail, we know that they are equal for S'_1 and different for S_1. We will use this observation in our side-channel attack on Keymill in the following.

3.2 Attack on Toy Model II

Basic Attack Strategy. For the sake of simplicity, we first describe the working principle of our attack on a slightly modified version of Toy Model II given in the Keymill specification [8], which has only two 8-bit shift-registers. In the attack, we assume that similar to Keymill, the output of the first flip-flop of each shift-register is not connected to the feedback function, as shown in Fig. 2. This is the only assumption that is necessary to mount our attacks. We do not rely on any other specific properties of the used feedback functions. The register is preinitialized with the secret key. After that, the 16-bit IV is absorbed, 2 bits per clock cycle. Our goal is to recover all *internal differences* of both registers after the IV (e.g., $IV = \texttt{0000}_{16}$) has been absorbed.

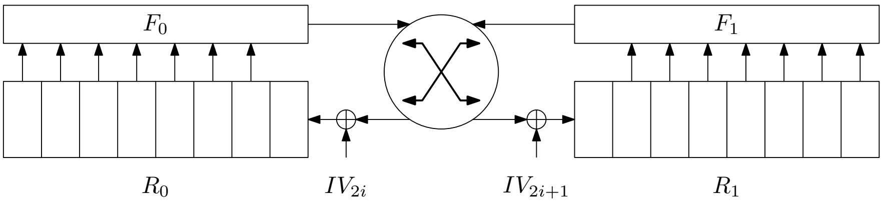

Fig. 2: Structure of modified Toy Model II

First of all, we collect two power traces, one for an IV starting with $\texttt{00}_2$ and one for an IV starting with $\texttt{10}_2$. We look at the power consumption when the first two bits are absorbed. Here, we have a difference in IV_0 for R_0, but equal values at IV_1 for R_1. Since the first flip-flop of each shift-register is not connected to the feedback function, the circuit processes the same information for both initial values, except for the first flip-flop of the left register R_0. As already discussed in Sect. 3.1, this gives us information about the difference of the first two bits of R_0 after absorbing the first two bits of the IV. If the power consumption when absorbing $\texttt{00}_2$ is higher than in the $\texttt{10}_2$ case, then we know that the first two bits of R_0 are different after $\texttt{00}_2$ is absorbed. If the power consumption is lower, then they are equal.

Next, we use two initial values starting with $\texttt{00}_2$ and $\texttt{01}_2$. This allows us to learn the internal difference of the first two bits of the register R_1 after $\texttt{00}_2$ is absorbed. Then, we use $\texttt{0000}_2$ and $\texttt{0010}_2$ to learn information of the difference of the first two bits after $\texttt{0000}_2$ has been absorbed, still preserving the information of the difference of the now second and third bits of both registers learned in the steps before. By continuing in this way, we can learn the differences of all neighboring bits of R_0 and R_1 after the IV $\texttt{0000}_{16}$ has been absorbed.

Now, guessing one bit in each register determines the other 7 bits. Hence, we are left with only 4 different possible internal states. From this state on, we can invert Toy Model II step by step until we recover the secret key.

3.3 Attack on Keymill

Compared to the Toy Model II, Keymill is just more of the same. Here, we have 4 registers, one 31-bit register, two 32-bit registers and one 33-bit register. The 128-bit IV is absorbed in 32 cycles, each cycle taking 4 bits. Hence, we can at most learn 32 differences of neighboring bits per register. Thus, we have to guess here in total a 4-bit information, giving us 16 different states leading to 16 key-candidates.

3.4 A Note on Filtering the Noise

The success of our attacks crucially depends on the ability to distinguish power consumption changes for a change of the input values. This means that the noise level has to be small enough to reliably identify these changes. If the attacker is allowed to repeat IVs, then averaging the trails and filtering the noise is no problem. Even if the IV is required to be unique, this can easily be done since the state of the registers only depends on bits of the IV that have already been absorbed. Hence, we can use all the remaining IV bits after the relation we want to recover to average the power consumption in this cycle. In this way, we can average over up to 16 power cycles even if we recover bit relations in the penultimate IV absorbing cycle.

4 Conclusion

In this work, we showed that a DPA on Keymill is feasible. In contrast to the DPA attacks that are claimed to be thwarted by the specification of Keymill [8], we do not recover the actual values of Keymill's internal state. Instead, we recover the differences of neighboring bits. Our attack violates the claim by the designers that Keymill is secure against SCA attacks inherently by design without requiring any redundant circuit. Indeed, we show that Keymill needs dedicated countermeasures against DPA attacks exploiting internal differences.

References

1. Burman, S., Mukhopadhyay, D., Veezhinathan, K.: LFSR based stream ciphers are vulnerable to power attacks. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) Progress in Cryptology – INDOCRYPT 2007. LNCS, vol. 4859, pp. 384–392. Springer (2007)

2. Clavier, C., Coron, J., Dabbous, N.: Differential power analysis in the presence of hardware countermeasures. In: Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2000. LNCS, vol. 1965, pp. 252–263. Springer (2000)
3. Gammel, B.M., Göttfert, R., Kniffler, O.: Achterbahn-128/80. eSTREAM, ECRYPT Stream Cipher Project (2006)
4. Herbst, C., Oswald, E., Mangard, S.: An AES smart card implementation resistant to power analysis attacks. In: Zhou, J., Yung, M., Bao, F. (eds.) Applied Cryptography and Network Security – ACNS 2006. LNCS, vol. 3989, pp. 239–252 (2006)
5. Medwed, M., Standaert, F.X., Großschädl, J., Regazzoni, F.: Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In: Bernstein, D.J., Lange, T. (eds.) Progress in Cryptology – AFRICACRYPT 2010. LNCS, vol. 6055, pp. 279–296. Springer (2010)
6. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of non-linear functions in the presence of glitches. In: Lee, P.J., Cheon, J.H. (eds.) Information Security and Cryptology – ICISC 2008. LNCS, vol. 5461, pp. 218–234. Springer (2008)
7. Prouff, E., Rivain, M.: Masking against side-channel attacks: A formal security proof. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. LNCS, vol. 7881, pp. 142–159. Springer (2013)
8. Taha, M., Reyhani-Masoleh, A., Schaumont, P.: Keymill: Side-channel resilient key generator. In: Avanzi, R., Heys, H. (eds.) Selected Areas in Cryptography – SAC 2016. LNCS, Springer (2016), `http://eprint.iacr.org/2016/710`
9. Zadeh, A.A., Heys, H.M.: Simple power analysis applied to nonlinear feedback shift registers. IET Information Security 8(3), 188–198 (2014)