Synchronous Sampling and Clock Recovery of Internal Oscillators for Side Channel Analysis

Colin O'Flynn and Zhizhang (David) Chen

Dalhousie University, Halifax, Canada
{coflynn, z.chen}@dal.ca

Abstract. Measuring power consumption for side-channel analysis typically uses an oscilloscope, which measures the data relative to an internal timebase. By synchronizing the sampling clock to the clock of the target device, the data storage and sampling requirements are considerably relaxed; the attack will succeed with a much lower sample rate. Previous work has demonstrated this on a system with a fixed and easily available clock; but real devices will often have an inaccessible internal oscillator, and may purposely vary the frequency this oscillator runs at (the Varying Clock countermeasure).

This work measures the performance of a synchronous sampling system attacking a modern microcontroller running a software AES implementation. This attack is characterized under three conditions: with a stable clock, with a clock that randomly varies between 4.5 MHz–12.7 MHz, and with an internal oscillator that randomly varies between 7.41 MHz–7.49 MHz.

Traces captured with the synchronous sampling technique can be processed with a standard Differential Power Analysis (DPA) style attack in all three cases, whereas when an oscilloscope is used only the stable oscillator setup is successful. This work also develops the required hardware to recover the internal clock of a device which does not have an externally available clock.

Keywords: side-channel analysis, acquisition, synchronization, DPA

1 Introduction

By measuring the power consumed by a digital device on each clock cycle, it is possible to infer something about the data being processed by this device. This was demonstrated as a method of breaking cryptographic cores using Differential Power Analysis (DPA)[1]. Such measurements are typically done with standard oscilloscopes, which depending on the attack algorithm and device under attack may range from simple low-cost oscilloscopes to high-end specialist oscilloscopes. But if the underlying objective is to measure data on the clock edges of the system clock, sampling at the clock rate of the system is sufficient, provided such samples occur at the correct moment (i.e. on the clock edge). This sampling technique is called synchronous sampling, where the sample clock is synchronized to the device clock. The application of this to side-channel analysis was described and demonstrated in [2], where the SASEBO-GII board was attacked, and this demonstrated how sampling at 96 MS/s synchronously achieved similar results to sampling at 2 GS/s asynchronously.

For this to be successful, the previous work assumed that the system clock was readily available. For many systems this will be the case—an external oscillator or clock drives the digital logic, and it is trivial to tap into this clock. But many devices rely instead on an internal oscillator; there is no clock signal available for synchronous sampling. In addition to the lack of the external clock source, devices may purposely vary the frequency of the internal oscillator in an attempt to stop power traces from synchronizing in the time domain, requiring the attacker to resynchronize the traces after capture.

This work addresses these two issues. First, an introduction to the reference platform being used is given, along with a comparison of the synchronous sampling technique to standard asynchronous sampling on this platform. The platform is then changed to use an internal oscillator which actively varies the frequency during cryptographic operations, and the performance of synchronous sampling is demonstrated. Finally a method of performing clock recovery, and using that clock for synchronous sampling is demonstrated.

2 Experimental Platform

The device under test (DUT) is an Atmel AtMega48A microcontroller in 28-pin DIP. This device was selected due to several clocking features: it can use an internal or external clock source, the internal oscillator can be adjusted by firmware running on the microcontroller during operation, and the internal clock can be output onto an I/O pin. The differential voltage is measured across a shunt inserted into the ground pin of the microcontroller. For asynchronous sampling an Agilent MSO 54831D oscilloscope is used, and for synchronous sampling the OpenADC is used along with a ZTEX FPGA board. Full details of the capture hardware and software are available in [2] and at the ChipWhisperer wiki[1]. See Fig. 9 for a photo of the test setup.

The 'A' suffix for the AtMega48A indicates it is using a recent fabrication process; the older AtMega48P by comparison is made with a larger (0.35μm) process. The AtMega48P draws more power, and thus would be expected to give a stronger signal across the resistive shunt used to measure current. The AtMega48A thus reflects a reasonable platform which can be compared against any recent digital IC.

The crypto module under attack is a C implementation of the AES-128 algorithm. The specific C implementation chosen was 'AES in C' available from avrcryptolib[2]. The attack algorithm is a standard Correlation Power Analysis (CPA) attack[3].

A comparison of asynchronous sampling and synchronous sampling is shown in Fig. 1. For this figure an external 7.37 MHz crystal oscillator was used as

[1] www.chipwhisperer.com

[2] http://avrcryptolib.das-labor.org

a clock source. Results in this paper will be an average of the partial guessing entropy (PGE) of all subkeys, and where space permits the PGE of each individual subkey is graphed. When the PGE is zero this means that each of the 16 subkeys were correctly detected. The expected PGE for a purely random distribution would be 128.

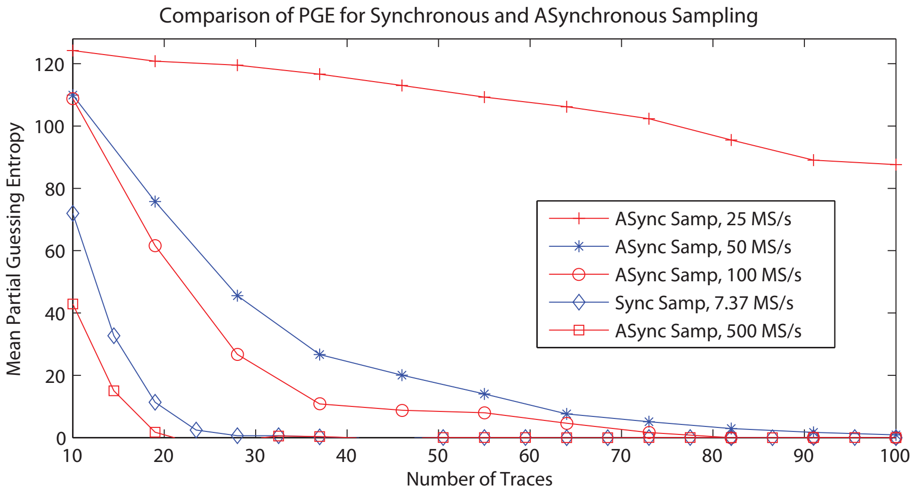


Fig. 1. The synchronous sampling method used here simplifies measurement work by allowing the use of a slower sampling speed, and thus shorter traces, with similar success rates to a high-speed oscilloscope.

3 Varying Clock Frequency

When an attacker is recording the power traces, ideally each trace would be perfectly synchronized with each other. That is to say that each time instance across all traces corresponds to the same instruction occurring on the DUT. In real systems, traces may not be perfectly synchronized. This could come from jitter in the trigger signal, unintended non-linear code flow such as interrupts on the DUT, or countermeasures such as instruction shuffling or random delay insertion. A discussion of algorithms and their performance for resynchronizing is compared in [4]. For all these events the clock is operating at a constant frequency.

Another class of synchronization aims to compensate for the clock frequency of the device varying (called *varying clock* or VC), either due to countermeasures or simply due to the oscillator drift. For an example of the natural variation see Fig. 2, which was measured the short-term drift of the internal oscillator on the experimental platform used here. This small amount of variance was enough to prevent the same CPA attack from being successful with over 2500 traces[3], when with a stable crystal oscillator it was successful in only 30 traces. Algorithms which aim to reverse the VC are given in [5–8].

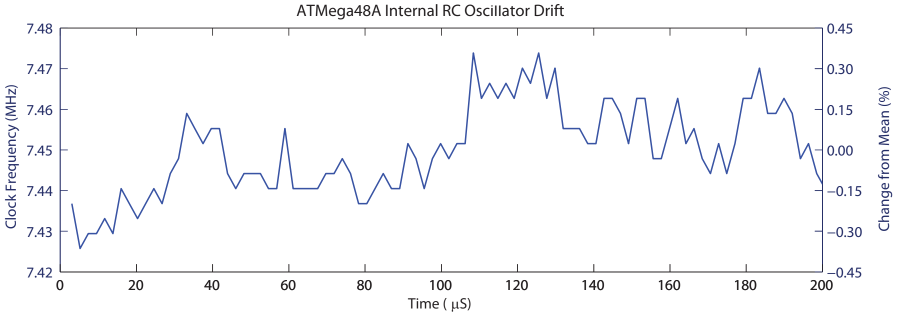


Fig. 2. Atmel AtMega48A internal clock drift during a side-channel attack.

With synchronous sampling, variations in clock frequency will naturally be eliminated from the data source. Each sample no longer corresponds to a time instant, but instead to a clock transition. Synchronization may be required for reasons previously discussed such as trigger jitter or countermeasures, but is not needed to compensate for the clock frequency changing.

3.1 Synchronous Sampling of Varying Clock

As a demonstration of synchronous sampling under VC conditions the AtMega48A target was designed to randomly vary the internal clock frequency before calling the AES encryption routines, and a side-channel attack was mounted. For this initial test the CLKOUT fuse was used to output the internal clock onto an IO pin, and the sampling is done synchronous to this clock.

The AtMega48A datasheet guarantees the oscillator can be calibrated between 7.3 MHz–8.1 MHz, but the actual range is much larger—the specific part used here had a range of 4.5 MHz–12.7 MHz. This test is operating the device outside of guaranteed operating range; commercial products would be advised to only use the adjustment over a smaller range. The time required to switch from the two possible extremes of the randomly selected frequencies, 4.5 MHz to 12.7 MHz, is shown in Fig. 3. The datasheet specifies a maximum change of 2% clock cycle period between cycles for an external clock; it is not clear if this rapidly changing internal oscillator would also be subject to these considerations[9]. For this reason a number of NOP instructs are inserted before beginning further processing after changing the OSCCAL register.

[3] After 2500 traces the average PGE was 40, and only 4 of the 16 bytes had a stable PGE < 5

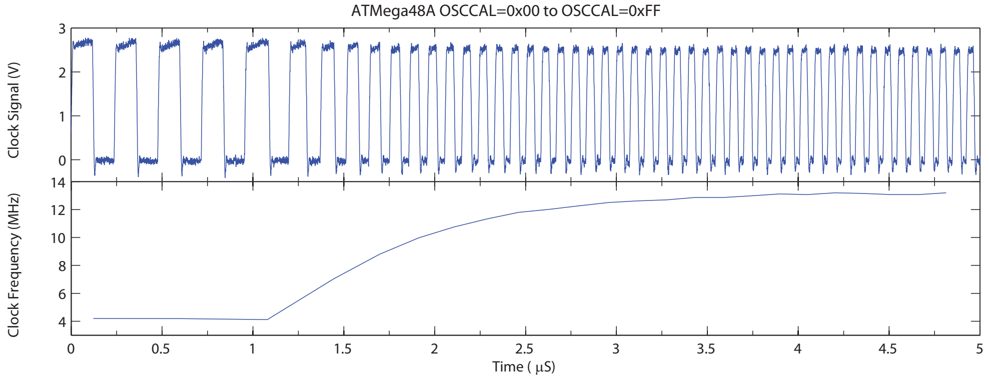


Fig. 3. Atmel AtMega48A internal clock frequency change as OSCCAL changes from 0 to 255.

3.2 Preprocessing of Traces

The power consumption of a digital device is dependent on the frequency of operation, and this follows a linear relationship. For the ATMega48A at 3.3V, the power consumption when moving from 4 MHz to 12 MHz goes from 1.7 mA to 3.1 mA[9]. While the power traces will line up in the time domain with synchronous sampling, they will require scaling in order to allow comparison of the same point across multiple traces. In [7] it is suggested to scale the traces based on the measured frequency of operation, as in (1). Here $T_{p,n}$ is a single point at index p in trace n, C is a scaling constant, and $f_{p,n}$ is the frequency of the clock at point $T_{p,n}$.

$$T'_{p,n} = T_{p,n} + Cf_{p,n} \tag{1}$$

The clock frequency was not captured alongside the traces in this work, and because there was no absolute time reference it could not be measured from the traces. Instead (2) is applied before passing onward to the CPA algorithm, which normalizes the traces by standard deviation.

$$T'_{p,n} = \frac{T_{p,n} - \mu_{T_n}}{\sigma_{T_n}} \tag{2}$$

Fig. 4 shows the traces before and after preprocessing—note the alignment in the time domain of all the traces due to synchronous sampling, despite the varying clock of the DUT.

Even with synchronous sampling, some trace resynchronization may be required. In this case if the sampling was started and then the clock speed changed, the traces had slight misalignment. It is assumed this comes from either the microcontroller delaying execution during the frequency change, or errors in the sampling ADC as the clock frequency changes. The synchronous sampling still greatly simplified the further resynchronization required, as all traces were within

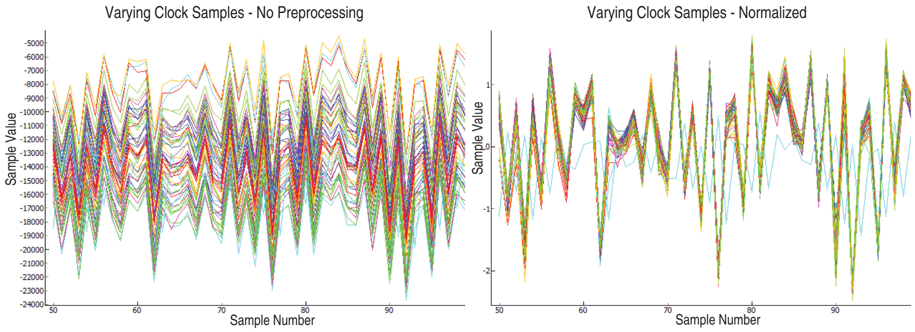


Fig. 4. Traces are made zero-mean and normalized by standard deviation before being passed on to the CPA attack.

3 samples (clock cycles) of each other. If the sampling was started after the clock frequency speed changed, no resynchronization was required, despite the DUT running at different frequencies.

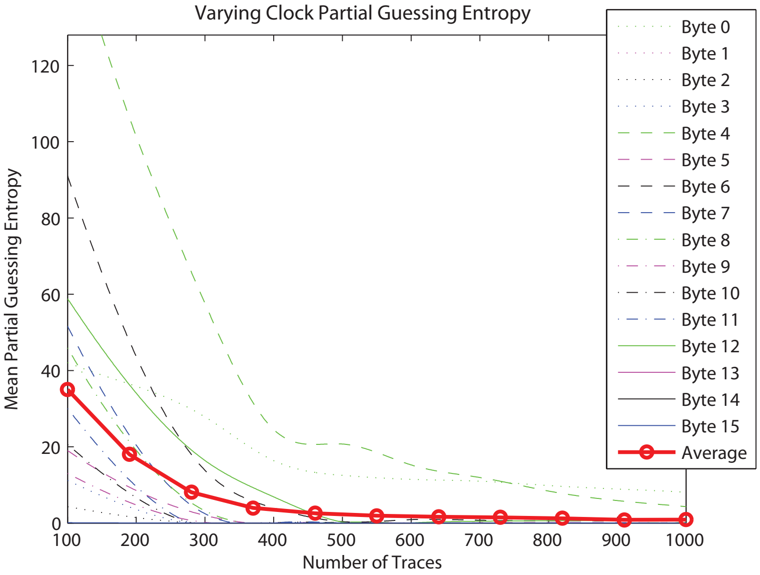


Fig. 5. Results of a CPA attack on a device with oscillator frequency randomly varying between 4.5 MHz–12.7 MHz on each encryption, and no trace synchronization being performed.

3.3 Results

The PGE of the CPA attack is shown in Fig. 5. While the PGE of the variable clock is worse than the PGE of the stable oscillator, the advantage of synchronous sampling is still demonstrated, that the CPA algorithm can be applied with no additional trace synchronization required. The clock frequency was not recorded for each trace due to current hardware limitations, meaning the correction factor described in [7] could not be applied, and instead normalization was used. Further work into attempting to use a correction factor based on measured frequency needs to be explored and compared to normalization. An additional method of compensating for the current variation with frequency is described in [8], this was not further explored as the normalization showed sufficiently good performance.

Beyond improving the correction factor, tests into attack algorithms which do not depend on the absolute peaks could be useful with synchronous sampling. Frequency-Based analysis such as at [10] could be used, where the synchronous sampling would have automatically scaled all frequency components to be relative to the master clock.

Considering the extremely large range the oscillator was varied over (4.5 MHz–12.7 MHz), these results do show that synchronous sampling is a viable method of attacking the varying clock (VC) countermeasure.

4 Clock Recovery

The previous section assumed that the clock was available for the synchronous sampling. In many devices the clock is not available externally, meaning additional work is required to perform this synchronous sampling. In side-channel analysis, it was previously demonstrated how to force an internal oscillator to lock to an external signal [11]. This was used to stabilize the internal RC oscillator and improve trace synchronization, but the same method could be used to generate the reference clock for synchronous sampling. This will fail if the device itself is varying the clock frequency, so instead *clock recovery* must be used to generate a copy of the clock. The idea of clock recovery is not new—in communications electronics this has been used for many years to synchronize a receiver clock to a transmitter clock over long distances[12].

The basic method used for clock recovery is to filter the power signal so that only the fundamental frequency from the internal oscillator is left. This can then be amplified and turned into a digital signal. To prevent glitches from resulting at the output a PLL is used to provide a clean digital signal. Details of this hardware design and results of side-channel analysis tests will be presented next.

4.1 Hardware Design

A block diagram of the system is given in Fig. 6, for a complete schematic see Appendix A. A Low Noise Amplifier (LNA) is placed on each side of the bandpass filter (BPF), the BPF selecting the fundamental frequency from the power signal. The output of the final LNA is limited to logic levels and fed into the Phase Lock Loop (PLL) block. The PLL used is a single-chip solution, the Texas Instruments CDCE906 device which integrates the Voltage Controller Oscillator (VCO), Phase Detect (PD), loop filters, and frequency dividers into a single package. For an introduction to PLLs the reader is referred to [13].

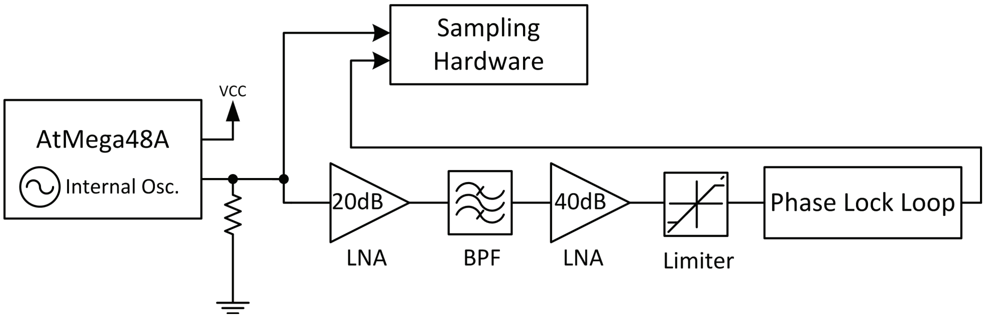


Fig. 6. Clock Recovery Block Diagram.

Fig. 7 shows an example of the unamplified signal, amplified but limited signal, and LVCMOS outputs. This specific example comes from attacking a KeeLoq transmitter's internal 1.3 MHz oscillator. The PLL has been disabled in this example to eliminate the lock time, typically 100μS[14]. Note the 'glitches' present in the output when the input signal disappears.

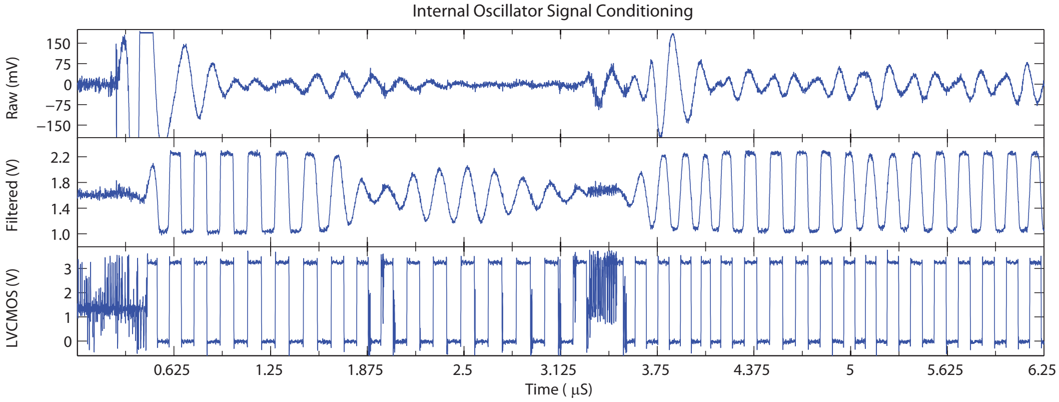


Fig. 7. Recovery of 1.3 MHz Internal RC Oscillator on KeeLoq single-chip hardware.

4.2 Filter Design

The design of the band-pass filter (BPF) is critical for the success of the clock recovery, details of the design process are given in Appendix A. Selection of the pass-band is based on the frequency of the internal oscillator for the device under attack. If this frequency is not known it can typically be found by viewing the frequency spectrum of the device during operation.

Careful consideration must be given for the group delay of the filter, which changes over frequency. As an example the 6.5 MHz–8.5 MHz BPF used for the ATMega48A device is shown in Fig. 8. The group delay, which is usually measured in time units, has been scaled by the frequency to give us a group delay in 'clock cycles'. The group delay will cause synchronization errors between traces if the frequency of the DUT oscillator changes, since the delay through the filter varies with frequency.

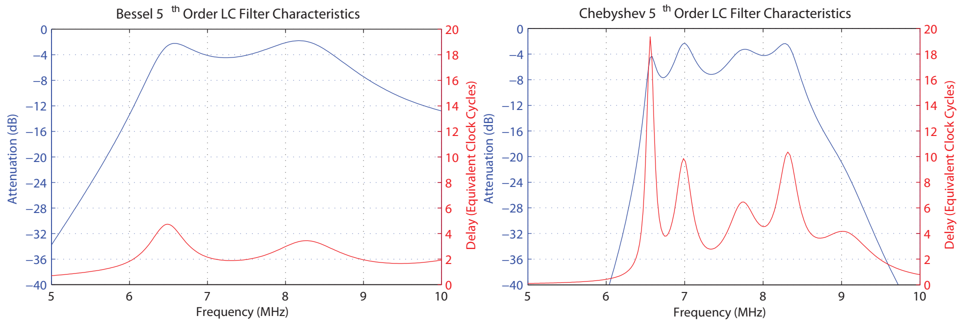


Fig. 8. Choice of filter type means a choice between better group delay performance and better attenuation outside the pass-band. Two examples are given here: a Chebyshev filter and a Bessel filter, both 5th order made from discrete LC components.

Three methods to reduce this error can be used. First, the type of analog BPF should be matched with the DUT. If the frequency of the oscillator varies only a tiny amount, it would be possible to use a Chebyshev filter with the better attenuation performance. If the DUT oscillator frequency will vary significantly a filter with better group delay performance could be used such as the Bessel. The second way to reduce this error is to measure the frequency during each trace acquisition, and shift the recorded waveform by the known group delay of the filter at this frequency. Finally a standard trace synchronization algorithm can be used to synchronize all such traces.

4.3 Results of CPA Attack

The AtMega48A platform is used again for this evaluation. The 'external clock' output is disabled during these tests—the AVR driving the IO pin at the clock frequency results in a very strong fundamental harmonic on the power trace, which results in a better signal for the PLL to lock onto. Such a system would be unrealistic since real systems would not be driving an arbitrary IO pin causing this strong fundamental.

The complete setup with clock recovery module, OpenADC capture hardware, and target is shown in Fig. 9.

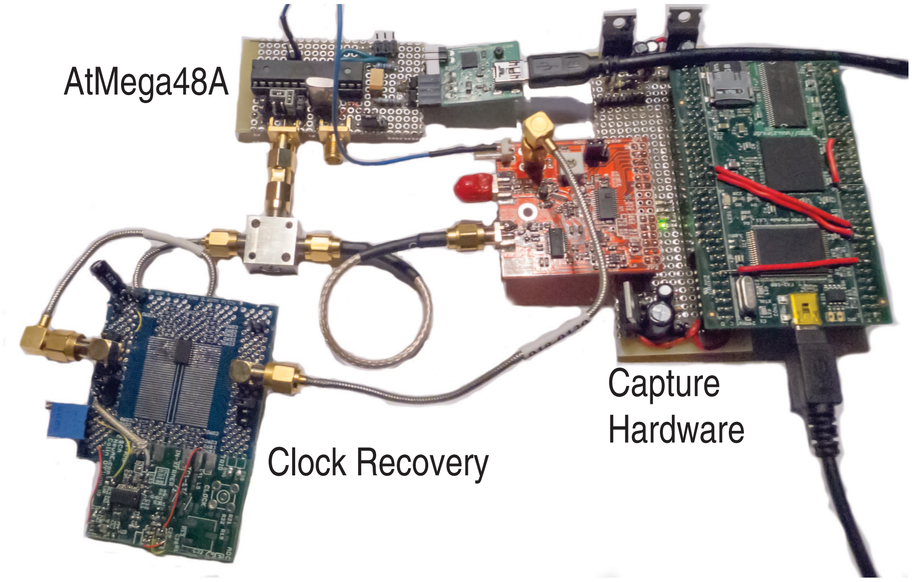


Fig. 9. Test Setup for side-channel analysis with clock recovery of internal oscillator on ATMega48A.

To avoid the problems with group delays of the filter, the variance of the internal oscillator was limited to ±0.5%. The center frequency was 7.45 MHz, choosen to correspond as close as possible with the reference crystal. The total range was thus 7.41 MHz–7.49 MHz. Fig. 10 compares the results of a CPA attack on traces captured with syncronous sampling using clock recovery, to a CPA attack on traces capture using a standard oscilloscope.

It can be seen the PGE of the attack on traces captured with synchronous sampling improves considerably, where the PGE of the attack on traces captured with an oscilloscope is only making modest improvements. The PGE does not progress to zero as rapidly as with a wired oscillator, and typically it is a few of the subkeys which are causing problems. As previously mentioned the group delay which varies with frequency is one problem which adds jitter to the traces. The clock recovery circuit itself also had some jitter in it causing the traces to not remain perfectly synchronized. Rather than sampling at the clock frequency, it may be beneficial to sample at a multiple of the clock frequency (e.g. 4x or 8x) and perform trace synchronization to account for some of this jitter.

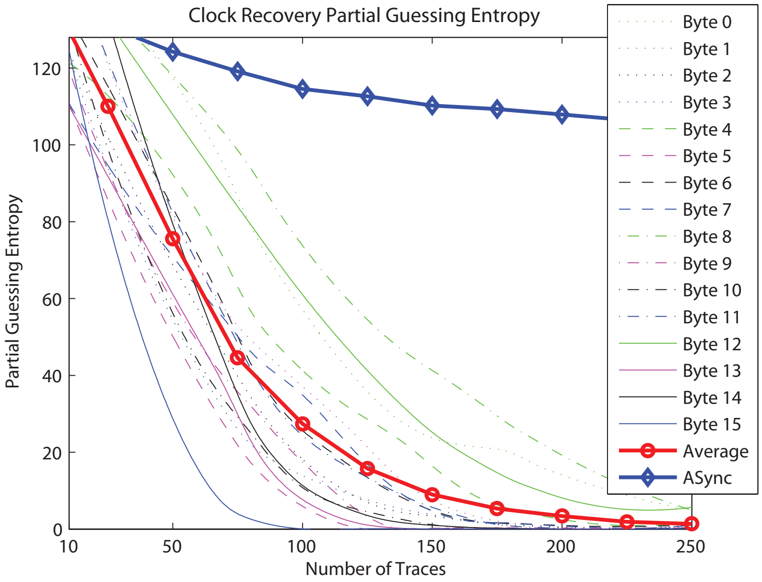


Fig. 10. Results of a CPA attack on a device with an internal RC oscillator, where the oscillator frequency changes during operation. The *Byte N* refer to the subkey Partial Guessing Entropy(PGE), *Average* refers to the average of all 16 subkeys, and *ASync* refers to the Average PGE for all subkeys of a CPA attack mounted on the same platform but using an oscilloscope sampling at 500 MS/s.

4.4 Further Work

Preprocessing to account for group delay in the filter may improve results, especially if the clock varies over a wider frequency range. In general this can be accomplished using standard trace synchronization schemes, which may also require sampling at a multiple of the clock frequency for increased time granularity.

Work on side-channel analysis of wireless devices has already been presented [15], using standard AM demodulating techniques. In a similar manner recovery of the clock from this waveform is also possible, and clock recovery may allow attacking more complex devices by providing a clock reference for synchronous sampling. There are many tightly integrated RF SoC devices, such as IEEE 802.15.4 radios, which integrate a microcontroller or AES hardware on the same die as a radio device. Careful design of a proper analog front-end, including the use of clock recovery, could make it possible to detect power variations modulated onto the RF carrier emitted by these devices.

5 Conclusions

Synchronous sampling has already been demonstrated to be a useful tool in reducing the data complexity when working with side-channel analysis measurements. It does this by reducing the data points to those of specific interest. When done via preprocessing this is known as *compression* of the power traces, but synchronous sampling eliminates the preprocessing requirement.

Synchronous sampling depends on the availability of the device clock, where many real devices contain an internal oscillator with no external signal. This paper has demonstrated how a ‘clock recovery’ technique can generate an external reference clock which is phase-locked to the internal oscillator of the device.

If the device under attack is varying the internal oscillator, this external clock will remain phase-locked to the true frequency. As synchronous sampling is measuring clock edges and not absolute time, this varying clock has very little effect of the success rate of an attack performed on these traces. The traces remain perfectly synchronized despite the changing clock frequency.

An interesting area of further work is taking traces captured with synchronous sampling and analyzing them in the frequency domain. The traces have effectively been scaled all by the clock frequency of the device, and thus should also ‘line up’ in the frequency domain despite the varying clock.

Acknowledgments Special thanks to funding provided by NSERC Canada Graduate Scholarship and OZ Optics.

References

1. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Advances in Cryptology - CRYPTO’ 99, Springer-Verlag (1999) 388–397
2. O’Flynn, C., Zhizhang, C.: A case study of Side-Channel Analysis using Decoupling Capacitor Power Measurement with the OpenADC. Lecture Notes in Computer Science **7743** (2013) 328–344
3. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. Cryptographic Hardware and Embedded Systems - CHES 2004 (2004) 135–152
4. Guilley, S., Khalfallah, K., Lomne, V., Danger, J.L.: Formal Framework for the Evaluation of Waveform Resynchronization Algorithms. In: Proceedings of the 5th IFIP WG 11.2 International Conference on Information Security Theory and Practice. WISTP’11, Berlin, Heidelberg, Springer-Verlag (2011) 100–115
5. van Woudenberg, J.G.J., Witteman, M.F., Bakker, B.: Improving Differential Power Analysis by Elastic Alignment. In: Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011. CT-RSA’11, Berlin, Heidelberg, Springer-Verlag (2011) 104–119
6. Kafi, M., Guilley, S., Marcello, S., Naccache, D.: Deconvolving Protected Signals. In: Availability, Reliability and Security, 2009. ARES ’09. International Conference on. (March 2009) 687 –694
7. Réal, D., Canovas, C., Clédière, J., Drissi, M., Valette, F.: Defeating Classical Hardware Countermeasures: A New Processing for Side Channel Analysis. In: Proceedings of the Conference on Design, Automation and Test in Europe. DATE ’08, New York, NY, USA, ACM (2008) 1274–1279
8. Tian, Q., Huss, S.: On Clock Frequency Effects in Side Channel Attacks of Symmetric Block Ciphers. In: New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on. (May 2012) 1 –5

9. Atmel Corporation: ATmega48A Datasheet
10. Gebotys, C.H., Ho, S., Tiu, C.C.: EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In: Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings. Volume 3659 of Lecture Notes in Computer Science., Springer (2005) 250–264
11. Skorobogatov, S.: Synchronization method for SCA and fault attacks. Journal of Cryptographic Engineering **1**(1) (April 2011) 71–77
12. Costas, J.: Synchronous Communications. Communications Systems, IRE Transactions on **5**(1) (March 1957) 99 –105
13. Banerjee, D.: PLL Performance Simulation and Design Handbook. 4th edn. Texas Instruments (2006)
14. Texas Instruments: CDCE906 Datasheet
15. Kasper, T., Oswald, D., Paar, C.: Side-channel Analysis of Cryptographic RFIDs with Analog Demodulation. In: Proceedings of the 7th international conference on RFID Security and Privacy. RFIDSec'11, Berlin, Heidelberg, Springer-Verlag (2012) 61–77

Appendix A: Hardware and Design Details

This appendix provides some brief notes on the physical hardware realized in this paper, along with a few notes for researchers looking to duplicate it.

5.1 Core Clock Recovery Module

The core part of this work is a module with a Low Noise Amplifier (LNA), Limiter, and Phase-Lock Loop (PLL) chip. The schematic for this is given in Fig. 11. The LNA is an Analog Devices AD8331, which has a variable gain up to 55dB. A resistor connected to the 'RLIM' pins provides an ability to set an arbitrary clipping level for the output. This clipped output is connected to the PLL chip, which is a Texas Instruments CDCE906. The clipped output from the LNA is used a LVDS input to the PLL, which works assuming the input to the entire block was sufficiently clean, that is to say contains only a single frequency component. Additional filtering can be added by placing capacitors on each of the input pins of the CDCE906 to ground, values between 100 pF–680 pF are reasonable depending on the fundamental frequency being targeted.

The CDCE906 was chosen for it's ability to operate down to 1 MHz, many PLL devices have higher lower frequency limits. If attacking devices with relatively slow internal oscillators, such as the KeeLoq devices at 1.3 MHz, this lower range is needed. The CDCE906 can be configured via I^{2}C to adjust parameters such as input drive level, frequency divider settings, and outputs in use. For this work it was configured to enable the PLL with frequency dividers such that the input and output frequency were the same. The sampling rate can easily be set to a higher multiple of the system frequency with this PLL block.

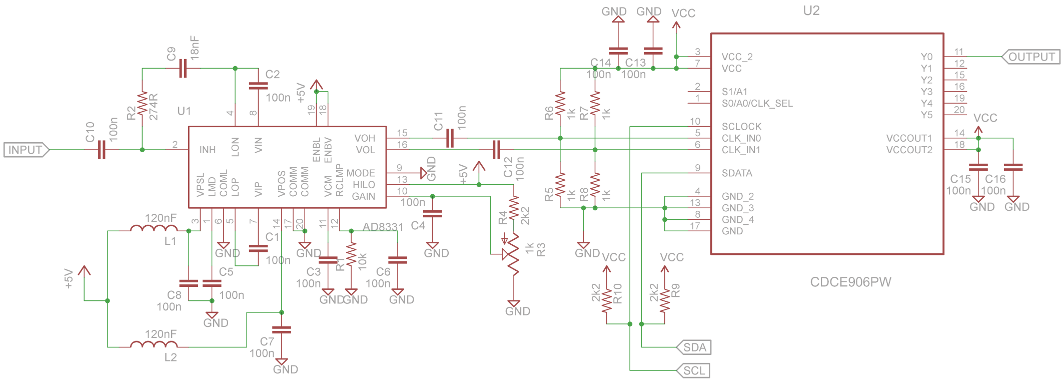


Fig. 11. Schematic for the LNA, Limiter, and PLL combined into one module from Fig. 6.

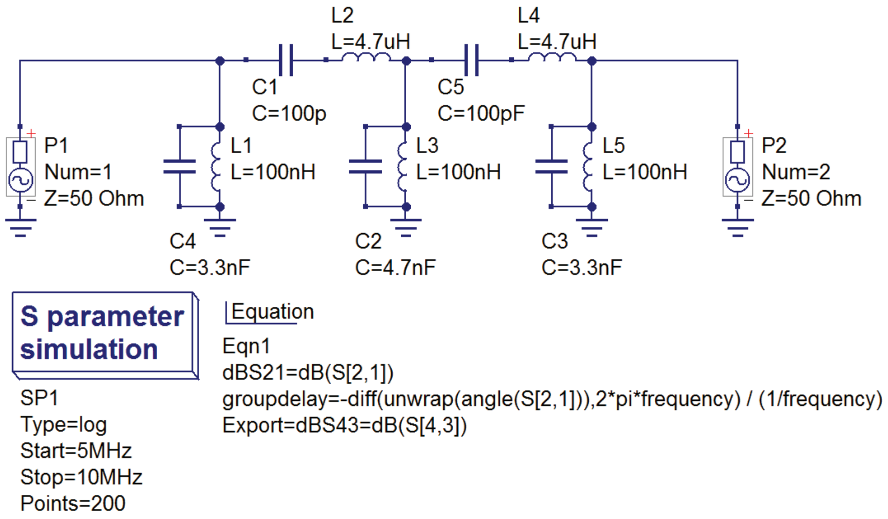
5.2 Filter

The filter design was done using the Quite Universal Circuit Simulator (QUCS) software. QUCS contains a *Filter Synthesis* tool, which can be used to generate an appropriate band-pass filter. This will be calculated with 'ideal' component values, and then these values are adjusted to the closest standard part, and a simulation confirms if the performance is still acceptable.

Note that at DC the filter will present a dead short, as no blocking capacitors are present. If connecting one side of the filter to a shunt or other device with a DC bias, always insert DC blocking capacitors.

5.3 First Stage LNA

An additional LNA may be required in front of the band-pass filter depending on the signal strength. It is possible to use a standard device such as a MiniCircuits *ZFL-1000LN+* which was used in this work. Care must be taken with RF amplifiers, as most of them are designed for use with 50Ω systems. If the output or input is not matched properly the amplifier may oscillate, causing errors. Generally amplifiers based on Op-Amps are safer in this regard, and specially-designed differential amplifiers can be exceedingly useful when measuring across current shunts.



```
SP1
Type=log
Start=5MHz
Stop=10MHz
Points=200
```

```
Eqn1
dBS21=dB(S[2,1])
groupdelay=-diff(unwrap(angle(S[2,1])),2*pi*frequency) / (1/frequency)
Export=dBS43=dB(S[4,3])
```

Fig. 12. Bandpass Filter Design Environment. Note the component values have been changed to reflect those being used in the actual circuit, and some optimizations may be needed to get acceptable performance. The equation to plot group delay in clock cycles can be seen in this diagram.