22nd Crypto-Day

Infineon Technologies AG
July 9 and 10, 2015
Munich, Germany

Visitor Information

Infineon Technologies AG – Neubiberg site (near Munich)

Dear Visitors,

Welcome to our Campeon site.

This leaflet outlines the various ways of traveling to Campeon, which is located just outside of Munich.

We look forward to seeing you!

- Please register at the Visitor Reception (Building 1) or at the Kubus Reception (Building 9).
- Please bring a photo ID or a driver‘s license.
- These can be exchanged for a Visitor ID, which you are required to wear visibly at all times.
- The receptionist will inform your contact person at your arrival. You will then be greeted by your contact person.
- You are not allowed to bring photo, video or audio devices with you to Campeon.
- If you wish to bring children along, you must obtain authorization from Infineon.
- Please be aware that entry into our premises is at your own risk.
- We ask for your understanding in the event of inspections by our security employees.

Contact

Campeon visitors’ address:
Infineon Technologies AG
Am Campeon 1-12
85579 Neubiberg
Germany

Tel. Reception:
+49 (89) 234 65400

Tel. Kubus Reception:
+49 (89) 234 65500

Infineon Service Center:
+49 (89) 234 0 (day & night)

Do you require a taxi to get to the airport or city center? Reception will be glad to help.

How to find us

Road

- A8 autobahn (highway), Exit 92b (Neubiberg), follow the signs to “Campeon”.
- At the “Campeon” roundabout take the second exit to the underground garage; registered visitors can contact the Reception via the button at the barrier and continue towards visitor parking (sign posted).
- Follow the signs to get from the underground garage to the Reception.

Visitors who arrive by taxis can be dropped off in front of the Visitor Reception (Building 1) or at the Kubus Reception.

S-Bahn

- Take the S3 train (from the center of Munich towards Holzkirchen / Deisenhofen) to “Fasanenpark” station (please do not confuse this station with “Fasangarten” station).
- Then take the covered walkway to Building 1 / Visitor Reception (approx. 3 minutes).

Rail

- 19 minutes from Munich Hauptbahnhof or
- 9 minutes from Munich Ostbahnhof with the S3 train to “Fasanenpark” station (see above).

Bus

- Bus 199 — From Neuperlach-Süd station (U5 and S7 trains) to either the “Campeon West” stop (KiTa) or the “Campeon” stop.
- Bus 220 — From St.-Quirinplatz station (U1 train) to the “Fasanenpark” stop.

Air

- From Munich Airport, take the S8 local train (S-Bahn) to Ostbahnhof (runs every 20 minutes).
- Then switch to the S3 local train towards Holzkirchen. Exit at “Fasanenpark” station.

Published by Infineon Technologies AG
Date: 02 / 2015

Here you find some information about the Crypto-Day together with the agenda for the two days and a list of participants. Please read especially the **information marked red** carefully.

Conference Location

The Crypto-Day is hosted by Infineon in Munich on the "Campeon", which is located in the south of Munich. Please see the attached leaflet on how to arrive at Campeon and where the buildings are located. The conference itself will be in building 02 in room 02.1.110/120.

You cannot enter the conference area individually; therefore, on both days, you need to join the group at the Visitor's Reception (see below) at the end of the registration phase (12:45 on Thursday, 8:45 on Friday).

Registration for access to Campeon

Before you enter the Campeon area, you will be required to register at the Visitor Reception (building 01). In order to register, you need to bring a photo ID (or driver's license) which you will be asked to leave at the reception in return for a visitor pass. The pass has to be worn visibly.

On Thursday 11:45-12:45 and Friday 8:15-8:45 we will be present in the Visitor Reception to make sure the registration process goes smoothly. On Thursday you are free to take an individual lunch after registration (see below), however, make sure to return to the Visitor Reception so we can go to the conference area together (see above).

You will need to register for each day individually, i.e., don't forget to return your visitor pass on Thursday evening.

Lunch in "Casino"

We will provide you with a "Gutschein" so you can enjoy the food served in our cantina. At Campeon we have two different cantinas in building 02 (same as conference building) on the ground floor: "Casino" (left side from entrance) and "Casino Plus" (right side). We have reserved a table in the back of the "Casino" marked with "Crypto-Day", but you are free to choose individual seats.

Your "Gutschein" is only valid for the left side, i.e., the regular "Casino".

Social Event

On Thursday evening, we plan to go to the "Augustiner Keller" beer garden which is located closely to the main station. The address of the beer garden is Arnulfstraße 52, 80335 München. Tramlines 16 and 17 stop in front of it (stop "Hackerbrücke" or "Hopfenstrasse"), but it is essentially walking distance from the station or any of the suggested hotels.

Accommodation

We recommend that you find accommodation in the general area of the Hauptbahnhof which is directly reachable by train and well connected to the airport. From there it is easy to reach Infineon (20 mins with Sbahn S3, see leaflet) and the beer garden of our social event.

Suggested hotels are the following:

1. **anna hotel**, Schützenstraße 1, 80335 München
2. **Hotel Condor**, Zweigstraße 6, 80336 München
3. **Helvetia Hotel**, Schillerstraße 6, 80336 München

Labtour

On Friday we offer a tour of Infineon's product security lab together with a talk, led by Dr. Peter Laackman. The tour itself will be 30 to 45 minutes long, the entire process will be longer due to a separate registration and individuals entering/exiting through a security door.

We can accommodate only 13 people, therefore you are required to send an email to benedikt.driessen@infineon.com, in case you want to join the tour. Participants will be chosen on a first-come first-serve basis. The deadline for this registration is 3.7.2015 at 10:00.

Agenda

Thursday, 9.7.		
1145	1245	**Registration and transfer to conference room**
1300	1320	Klimke: *Welcome & Keynote*
1320	1340	Kresmer, De Santis, Seuschek, Heyszl, Sigl: *High-Speed Curve25519 Scalar Multiplication on ARM Cortex-M4 32-bit Microcontrollers*
1340	1400	Boehm, Bucci, Hofer, Luzzi: *A Reliable Low-area Low-power PUF-based Key Generator*
1400	1430	**Break**
1430	1450	Zenger, Paar: *Channel-based Key Extraction for Wireless Resource-constrained Devices*
1450	1510	Burlakov, vom Dorp, von zur Gathen, Hillmann, Link, Loebenberger, Lühr, Schneider, Zemank: *Comparative analysis of pseudorandom generators*
1510	1530	Wild, Moradi, Güneysu: *GliFreD - Glitch-Free Duplication*
1530	1600	**Break**
1600	1620	Dinur, Dunkelman, Kranz, Leander: *Integral Attack on the ASASA Block Cipher Construction*
1620	1640	De Santis, Bauer, Sigl: *Higher-Order Polynomial Masking Hardware Implementations of AES*
1640	1700	Mikhalev, Armknecht: *On the Design of Stream Ciphers with Shorter Internal State*
1700	1720	Lesjak, Hein, Winter: *Red/Green - Hardware-Security Technologies for Internet-of-Things (IoT)*
1900		**Beergarden (Augustiner Keller, Arnulfstr. 52)**

Friday, 10.7.		
0815	0845	**Registration and transfer to conference room**
0900	0920	Kiss, Krämer: *Self-Secure Exponentiation Countermeasures Against Fault Attacks and Power Analysis for RSA-CRT*
0920	0940	Dobraunig, Eichsleder, Mendel, Schläffer: *ASCON - Submission to the CAESAR Competition*
0940	1000	Sasdrich: *Side-Channel Protection with Dynamic Logic Reconfiguration and Randomized Look-Up Tables on FPGAs*

1000	1020	Druml, Schilling, Pachler, Roitner, Ruiprechter, Bock, Holweg: *Towards Miniaturized System-in-Package contactless and Passive Authentication Devices featuring NFC*
1020	1050	**Break (incl. chance to talk to Infineon HR about vacancies)**
1050	1110	Seuschek, Heyszl, De Santis: *Side-Channel Implications of Deterministic DSA-Signature Variants*
1110	1130	Günther: *Implementing Cryptographic Pairings on Infineon SLE 78*
1130	1150	Beierle: *Bounding the Differential Probability of SIMON*
1150	1210	Buescher, Katzenbeisser: *Faster Secure Computation through Automatic Parallelization*
1210	1330	**Joint lunch in "Casino"**
1330	1500	**Labtour**

Participants

Last Name	First Name	Affiliation
Armknecht	Frederik	Uni Mannheim
Ben Romdhane	Molka	Infineon
Beierle	Christof	Uni Bochum
Boehm	Christoph	Infineon
Buescher	Niklas	TU Darmstadt
Burlakov	Aleksei	Uni Bonn
De Santis	Fabrizio	TU München
Driessen	Benedikt	Infineon
Druml	Norbert	Infineon
Günther	Peter	Uni Paderborn
Heyszl	Johann	Fraunhofer AISEC
Issovits	Wolfgang	Infineon
Kiss	Agnes	TU Darmstadt
Kranz	Thorsten	Uni Bochum
Leicher	Florian	Uni Mannheim
Lesjak	Christian	Infineon
Lühr	Jan	Uni Bonn
Mikhalev	Vasily	Uni Mannheim
Müller	Christian	Uni Mannheim
Reuter	Christian	Uni Mannheim
Sasdrich	Pascal	Uni Bochum
Schilling	Jürgen	Infineon
Seuscheck	Herrmann	TU München
Schlaeffer	Martin	Infineon
Schneider	Simon	Uni Bonn
Tempelmeier	Michael	TU München
vom Dorp	Johannes	Uni Bonn
Wild	Alexander	Uni Bochum
Zenger	Christian	Uni Bochum

High-Speed Curve25519 Scalar Multiplication on ARM Cortex-M4 32-bit Microcontrollers

Patrick Kresmer*, Fabrizio De Santis*, Hermann Seuschek*, Johann Heyszl† and Georg Sigl*,†

*Technische Universität München
Munich, Germany
{patrick.kresmer,desantis}@tum.de
{hermann.seuschek,sigl}@tum.de

† Fraunhofer Institute AISEC
Munich, Germany
johann.heyszl@aisec.fraunhofer.de
georg.sigl@aisec.fraunhofer.de

Curve25519 is a 128-bit secure elliptic curve introduced by Daniel J. Bernstein in 2006 [BJD06] for use with the elliptic curve Diffie-Hellman (ECDH) key agreement scheme, which has received increasing attention in the past few years from both academia and the industry.

Very recently, the Curve25519-based ECDH has been implemented on a variety of embedded systems such as AVR ATMega 8-bit, MSP430 16-bit and ARM Cortex-M0 32-bit microcontrollers achieving particularly relevant speed results [DHH15].

In this work, we take a step forward implementing the Curve25519 scalar multiplication on ARM Cortex-M4 microcontrollers for different multiplication algorithms. In particular, we achieve improved constant-time speed records by taking advantage of single-cycle multiply-and-accumulate (MAC) instructions as available on ARM Cortex-M4 processors.

References

[BJD06] Daniel J. Bernstein. Curve25519: new Diffie-Hellman speed records. *Public Key Cryptography*, Springer, 207–228, 2006.

[DHH15] Michael Düll and Björn Haase and Gesine Hinterwälder and Michael Hutter and Christof Paar and Ana Helena Sánchez and Peter Schwabe. High-speed Curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. https://eprint.iacr.org/2015/343.pdf, April 2015.

A Reliable Low-area Low-power PUF-based Key Generator

Christoph Boehm, Marco Bucci, Maximilian Hofer and Raimondo Luzzi

Infineon Technologies Austria AG
{christoph.boehm, marco.bucci, maximiliam.hofer, raimondo.luzzi}@infineon.com

Physically Unclonable Functions (PUFs) are gaining more and more attention as a primitive function in the field of hardware security. Several applications have been proposed in literature: identification and authentication, IP protection and software-hardware binding, secure generation/storage of cryptographic keys. In spite of the wide range of contributions in this field, a complete characterization (under process and environmental variations) of the proposed ideas is not always available. Often too few devices are tested or temperature variations are not taken into account.

In this work, we focused on the usage of PUFs as secure key generators and the target was the design of a reliable small footprint PUF module which can be used as key generator in a chip-card controller, as a replacement or in addition to a key stored in a non-volatile memory (NVM). Since area and power consumption are main constraints in a chip-card controller, the focus was on the design of a custom PUF cell which is inherently more reliable than a standard latch or SRAM cell, thus reducing the complexity of the error correction scheme. The proposed two stages identification (TSID) cell operates in two phases behaving as a differential amplifier during a firrst phase, in order to amplify the local mismatch of two minimum area transistors, and as a latch as soon as a trigger signal is activated. In addition, in order to sort out the few cells that, due to the little mismatch, are more sensitive to noise, temperature and parameter variations, the pre-selection technique introduced in [HB10] has been adopted. Mask data are generated before the error correction code (ECC) calculation during the enrollment and stored in the NVM. During the key reconstruction, the raw data from the PUF cell array (PCA) are first compacted by applying the mask data and then the error correction is performed. The PCA consists of 1056 TSID cells organized in 22 blocks of 48 cells each. The 48 cells in a block share the same sense amplifier (SA), thus strongly reducing the area of the array. The module has been integrated into a chip-card controller and manufactured in a 65nm CMOS process. The PUF cell array shows a power consumption per bit of 4.2W at 100MHz with an area per bit of $2.4\mu m^2$.

Extensive tests have been performed on the raw data by measuring, over temperature, about 100 devices from different lots. The masking functionality has been tested, by varying the amount of pre-selection up the the maximum value which still leaves enough cells for the ECC (up to 7 bits in each 22 bit block can be discarded). Afterward, instability (i.e. the cumulative number of bits which are not stable over the performed readouts), inter-and intra-chip Hamming distance (HD), bias and spatial correlations have been measured, finding out that, over the temperature range -40/+110°C, an instability of about 11% and 2% has to be expected for the proposed PUF before and after pre-selection respectively. The Intra-chip HD (i.e. BER) is below 5% without pre-selection and drops down to 0.7% (at -40°C), if pre-selection is applied. Finally, the stability of the PUF key has been tested by performing the enrollment at 25°C and 10^6 key reconstructions at -40°C and +110°C respectively. After testing 3 wafers from different process corners, for a total of about 2000 devices, no single key reconstruction fail has been detected.

References

[HB10] M. Hofer and C. Boehm. An Alternative to Error Correction for SRAM-Like PUFs. *Proc. Workshop on Cryptographic Hardware and Embedded Systems*, LNCS, vol. 6225, 644–654, 2010.

Channel-based Key Extraction for Wireless Resource-constrained Devices

Christian T. Zenger and Christof Paar

Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
{christian.zenger,christof.paar}@rub.de
phone: +49 (0)234 3226533, http://www.hgi.rub.de

Abstract. Using the properties of a wireless channel is an alternative approach for securing the channel besides pre-shared keys or asymmetric cryptography. Numerous experiments have recently demonstrated that channel-based key establishment (CBKE) is a promising alternative to well-known symmetric/asymmetric approaches. Their run-times for establishing a symmetric key suggest that such methods are highly suitable for real-world applications that operate in a dynamic mobile environment with peer-to-peer association.

CBKE is a new paradigm for generating shared secret keys. The approach is based on the estimation of the wireless transmission channel by both the sender and receiver, where the shared secret key is derived from channel parameters. The *commonness* of the randomness of the secret key relies on the principle of *channel reciprocity*. Specifically, this means that the channel from Alice to Bob is the same than the channel from Bob to Alice. This symmetry of practical channels is usually sufficiently high, as well as its entropy of spatial, temporal, and spectral characteristics.

Security is given if an attacker's distance to the two communicating nodes is high enough, so that the observed channel parameters to each node are uncorrelated and independent from each other. Typically, in real environments this is given if the distance is greater than about half of the carrier wavelength. For instance, for the frequency used in 2.4 GHz WiFi, this translates to a distance of 6.25 cm.

So far, high usability and dynamic key management are very difficult to achieve for wireless devices, which operate under strict resource constraints. CBKE has the potential to significantly reduce the cost of securing small embedded devices, and hence make mass production and deployment more viable. Until now, no research has addressed the requirements for performance evaluation of real-world implementations of CBKE systems.

We present a wireless CBKE security system built with standard components, e.g., quantization scheme and error correction codes, presented in recent publications. We introduce necessary implementation properties and requirements of CBKE systems. In order to validate the performance of the key generation algorithms, we define a set of metrics. Finally, we describe an end-to-end implementation on an ARM-Cortex M3 microcontroller to demonstrate the practical feasibility of channel-based key estimation using current embedded hardware.

Comparative analysis of pseudorandom generators

Aleksei Burlakov, Johannes vom Dorp, Joachim von zur Gathen, Sarah Hillmann, Michael Link, Daniel Loebenberger, Jan Lühr, Simon Schneider & Sven Zemanek

{burlakov,dorp,luehr,schneid,zemanek}@cs.uni-bonn.de
{sarah.hillmann,michael.link}@uni-bonn.de
{gathen,daniel}@bit.uni-bonn.de
Bonn-Aachen International Center for Information Technology
Dahlmannstr. 2, Bonn

We compare random generators (RGs) under controlled conditions regarding their efficiency and statistical properties. For this purpose, we distinguish between physical RGs and software RGs, which can be further subdivided into cryptographically secure and insecure RGs.

Physical RGs covered by our study are the hardware generator PRG310-4 and `/dev/random` as implemented in the Linux kernel. Since `/dev/random` is fed by system events, we analyze both an idle lab environment and a server hosting several virtual machines. As examples for cryptographically secure RGs our analysis compares the RSA generator and the Blum-Blum-Shub generator, both for 3000-bit moduli. Additionally, we compare them to the Nisan-Wigderson construction with suitably selected parameters. We include two cryptographically insecure RGs, namely a linear congruential generator (LCG) and the Littlewood generator.

In order to obtain repeatable and comparable results, our implementations of the software RGs were all run on the same machine and produced 512 kB of output each, using AES post-processed output of the generator PRG310-4 as source for random seed bits. We compare the results in terms of byte entropy and throughput *excluding initialization*. For further statistical analysis — not shown in the table — we apply the NIST test suite on the outputs.

The most important finding is that in our scenarios, number-theoretic generators compete very well against hardware-based ones.

	byte entropy [bit]	runtime [μs]	throughput [kB/s]	throughput normalized
PRG310-4, no post-processing	7.99963	16308400	31.39486	4.34492
AES post-processing	7.99963	36524300	14.01806	1.94004
`/dev/random`, in the field	7.99979	9.169×10^{10}	5.584×10^{-3}	7.728×10^{-4}
in the lab	7.99948	2.671×10^{12}	1.917×10^{-4}	2.653×10^{-5}
Littlewood	6.47244	15206550	33.66970	4.61011
Linear congruential generator	7.99969	2644039	193.64313	26.51392
Blum-Blum-Shub	7.99962	17708350	28.91291	3.95880
RSA, $e = 2^{16} + 1$, 1400 bit/round	7.99966	267604	1913.27484	261.96857
$e = 3$, 1 bit/round	7.99963	70103838	7.30345	1
Nisan-Wigderson	7.99961	2731227	187.46153	25.66753

Table 1: Overview of the results for generating 512 kB of output.

1 Introduction

A crucial ingredient of almost every commonly used cryptographic scheme is the internal choice of random bits for key generation. This includes the unpredictability and hence the absence of statistical regularities within the random sequence of output. Doing so is not trivial.

One approach is to use system events (keyboard, mouse, etc.) for gathering entropy. This is done on Linux systems and made available via the devices `/dev/random` and `/dev/urandom`. While `/dev/random` blocks if no entropy is available on the system, `/dev/urandom` uses a pseudorandom generator (PRG) to generate output, see Gutterman, Pinkas & Reinman (2006); Lacharme, Röck, Strubel & Videau (2012).

Having enough entropy and randomness is *vital* for any kind of key generation — especially for systems relying on Diffie-Hellman key exchange protocols to achieve perfect forwarding secrecy. For instance, not having enough randomness causes TLS based online banking to stop and emails to be held back. Thus, modern computer systems desperately need cryptographically secure PRGs and hardware entropy sources, delivering seed entropy to PRGs or key generators with very high speed.

This paper provides a comparative analysis of some popular pseudorandom generators and two physical sources of randomness. As source of random seeds we use the PRG310-4 Concerning PRGs, we discuss the RSA generator, the Blum-Blum-Shub generator and the Nisan-Wigderson construction, which come with formal security proofs, as well as the Littlewood generator and the linear congruential generator as examples of insecure generators in the context of cryptographic applications.

Therefore, we first present previous work in the field of generator analysis in section 2 before giving a detailed overview of the generators in section 3. The main contribution will be the evaluation regarding efficiency and statistical uniformity in section 4, followed by a short conclusion in section 5. The last section 6 gives some hints on future work, building on the findings of this paper.

All algorithms were implemented in a textbook manner using non-optimized C-code, thus providing a fair comparison. The source code of the algorithms is available at Burlakov, vom Dorp, Hillmann, Link, Lühr, Schneider & Zemanek (2015). This paper is the result of a class project in a course taught by Joachim von zur Gathen and Daniel Loebenberger.

2 Related work

Concerning the physical generators, Schindler & Killmann (2003) analyzed noisy diodes as a random source, proving a model for its entropy. One example of a noisy diodes based generator is the generator PRG310-4, which is distributed by Bergmann (2014). Linux' VirtIO generator as used in `/dev/random` is illustrated by Gutterman *et al.* (2006) and explained by Lacharme *et al.* (2012). Both provide a clear picture of its inner workings.

Referring to PRGs, the RSA based generator is explained in Shamir (1983); Fischlin & Schnorr (2000); Steinfeld, Pieprzyk & Wang (2006). Its cryptographic security is shown in Alexi, Chor, Goldreich & Schnorr (1988) and extended in Steinfeld *et al.* (2006).

Linear congruential generators (LCGs) were first proposed by Lehmer (1951). Attacks were discussed in Boyar (1989); Plumstead (1982); Hastad & Shamir (1985). They all exploited its simple linear structure and come with a specific parameterization. Not all parameterizations — such as truncating its output to a single bit — have been attacked successfully as of today.

Blum, Blum & Shub (1986) introduced the Blum-Blum-Shub generator. Alexi *et al.* (1988) and

Fischlin & Schnorr (2000) showed that the integer modulus can be factored, given a break of the generator.

The generator by Nisan & Wigderson (1994) is proven to be cryptographically secure against constant depth circuits.

In the chapter "Mathematics with Minimum Raw Materials" of his book "A Mathematicians Miscellany", Littlewood (1953) proposes what is today called a stream cipher. This construction can be interpreted as a random generator.

We are not aware of any comprehensive benchmarking survey for these generators that integrates them into the Linux operating system.

3 The generators

In the following, each generator which was implemented or applied for the comparative analysis is briefly presented. The output of a pseudorandom generator is, by definition, not efficiently distinguishable from uniform randomness. Under suitable yet reasonable assumptions, the Blum-Blum-Shub, RSA, and Nisan-Wigderson generators have this property, but the LCG with full output and the Littlewood construction definitely do not.

3.1 Linux `/dev/random`

As already mentioned system events are used to gather entropy on Linux Systems. Lacharme *et al.* (2012) explains:

> "[Linux] processes events from different entropy sources, namely user inputs (such as keyboard and mouse movements), disk timings and interrupt timings."

These events are post-processed and made available to `/dev/random` and `/dev/urandom`. This includes estimating the entropy of the event and mixing.

The *Bundesamt für Sicherheit in der Informationstechnik* (BSI) sets cryptographic standards in Germany and judges both `/dev/random` and `/dev/urandom` to fulfill the requirements for their class NTG.1, except for NTG.1.1 which is not satisfied by `/dev/urandom`, see Müller, Krummeck & Romsy (2014). The definitions of the classes for random number generation can be found in Killmann & Schindler (2011).

3.2 PRG310-4

The PRG310-4 gathers entropy from a system of two noisy diodes, see Bergmann (2014), and is connected to a computer via USB. Similar variants exist for different interface types.

According to Bergmann (2014), its behavior follows the stochastical model in Schindler & Killmann (2003), who show that

> "[...] principally, this RG [design] could generate up to 700000 random bits per second with an average entropy $1 - 10^{-5}$."

Bergmann (2014) mentions that this device satisfies all requirements for PTG.3 class random generation.

3.3 The Littlewood generator

The mathematician E. J. Littlewood suggests a stream cipher based on properties of the logarithm function. When translated from a 26-ary alphabet to bits, this yields the following production rule: For a starting value x and fixed d, the ith bit of the key stream is the dth post-decimal bit of $\log_2(x+i)$. Littlewood writes about the cipher:

> "The legend that every cipher is breakable is of course absurd, though still widespread among people who should know better. I give a sufficient example, without troubling about its precise degree of practicability. [...] It is sufficiently obvious that a *single* message cannot be unscrambled [...]"

It has been shown by Wilson (1979) and Stehlé (2004) how Littlewood's cipher can easily be broken. As the output bits of the generator are taken from a curve over the reals, consecutive output bits can be used to recover this curve, and therefore the seed that is used to initialize the generator.

Moreover, the statistical properties of the generator can be further compromised by improper parameter choice. Call $\log_2(x+i)$ the $x+i$-th *row value*. When $x+i$ approaches a power of 2, the row values approach a whole number, and the first post-decimal bits of them will be 1. Littlewood addresses this implicitly, by taking the 6th and 7th post-decimal digits of a logarithm with 5 input digits. In this way, he avoids the first five digits, which predictably approach the value .99999. When the generator is expected to produce a maximum of N output bits, the first $\lceil \log_2(x+N) \rceil$ post-decimal bits of the row values have an elevated probability of being 1. Thus, we need to have d greater than that in order to not reduce the bit entropy unnecessarily.

3.4 Linear Congruential generators

Linear congruential generators as presented in Lehmer (1951) produce a sequence of values in $\mathbb{Z}_M$, generated by iteratively applying

$$x_i = s \cdot x_{i-1} + t \text{ in } \mathbb{Z}_M$$

to a secret random seed $x_0 \in \mathbb{Z}_M$ provided by an external source. The secret parameters s and t are chosen from $\mathbb{Z}_M$.

While the byte distribution of LCG outputs is generally well-distributed, with byte entropy close to maximal, the generated sequences are predictable and therefore cryptographically insecure. The author of the two papers Plumstead (1982) and Boyar (1989) describes how to recover the secret (s,t,M) from the sequence of $(x_i)_{i\geq 0}$ alone. A possible mitigation against this attack is to output only some *least significant bits* of the x_i. Håstad & Shamir (1985) describe a lattice based attack on truncated LCGs where (s,t,M) are public. This attack can be used to predict LCGs that output at least $\frac{1}{3}$ of the bits of the x_i. We are neither aware of a more powerful attack on the LCG nor of a security argument for a certain application.

In our evaluation we varied truncation lengths, without observing significant changes in the statistical features of the output. As a conservative measure, we decided to publish results for an LCG outputting only the least significant byte of every x_i, which seems reasonably secure.

In our implementation, M was to chosen to be the next prime, given a 3000 bit seed. By doing so, primality testing increases initialization and cold start times.

3.5 The Blum-Blum-Shub Generator

The Blum-Blum-Shub generator produces a pseudorandom bit sequence from a random seed by repeatedly squaring modulo a so called *Blum integer* $N = p \cdot q$, where p and q are large random primes congruent to 3 mod 4. In each round, the least significant bit of the intermediate result is returned.

Alexi *et al.* (1988) proved that factoring the Blum integer N can be reduced to being able to guess the least significant bit of any intermediate square with nonnegligible advantage. The output of this generator is thus cryptographically secure under the assumption that factoring Blum integers is a hard problem.

3.6 The RSA generator

The RSA generator was first presented by Shamir (1983) and is one of the PRGs that are proven to be cryptographically secure under certain number-theoretical assumptions. Analogously to the RSA crypto scheme, the generator is initialized by choosing a modulus N as the product of two large random primes, and an exponent e with $1 < e < \varphi(N) - 1$ and $\gcd(e, \varphi(N))$. Here, φ denotes Euler's totient function.

Starting from a seed x_0 provided by an external source the generator iteratively computes

$$x_{i+1} = x_i^e \mod N,$$

extracts the least significant k bits of each intermediate result and concatenates them as output.

Our implementation uses a random 3000-bit Blum integer (see section 3.5) as modulus N and various choices for the parameters e and k. For our tests, we choose $e = 3$, provided that $\gcd(e, \varphi(N)) = 1$, as for small exponents the generator is expected to work fast and because it allows us to compare the results to the runtime of the Blum-Blum-Shub generator. Additionally, we test the larger exponent $e = 2^{16} + 1$, which is widely used in practice for it is a prime and its structure allows efficient exponentiation.

Alexi *et al.* (1988) it is shown that the RSA generator is pseudorandom for $k \leq \log n$, under the assumption that the RSA inversion problem is hard. Here, n denotes the bit size of the modulus. Under stronger assumptions regarding the hardness of the RSA inversion problem, Steinfeld *et al.* (2006) prove the security of the generator for $k \leq n \cdot (\frac{1}{2} - \frac{1}{e} - \varepsilon - o(1))$ for any $\varepsilon > 0$. We test the two choices of $k = 1400$ and $k = 1$, the latter for comparison with the BBS generator.

3.7 The Nisan-Wigderson generator

Each output bit of the Nisan-Wigderson PRG is produced by a "hard" function $f : \{0,1\}^s \to \{0,1\}$. The arguments of f are chosen sequentially from a design $S = (S_1, S_2, S_3, ..., S_n)$ of subsets of $\{1, \ldots, k\}$ for some $k \in \mathbb{N}$, which can intersect in at most t elements, so that for all $i \neq j \leq n$ we have

$$S_i, S_j \subset \{1, ..., k\},\ s = \#S_i = \#S_j, \#(S_i \cap S_j) \leq t.$$

The output y is generated by applying f to arguments derived from a k-bit seed x. Each set $S_i = \{S_i^1, \ldots S_i^s\}$ $(1 \leq i \leq n)$ describes the bits of x at which f is evaluated. For the ith output bit y^i in y we have

$$y^i = f(x|_{S_i^1, ..., S_i^s}).$$

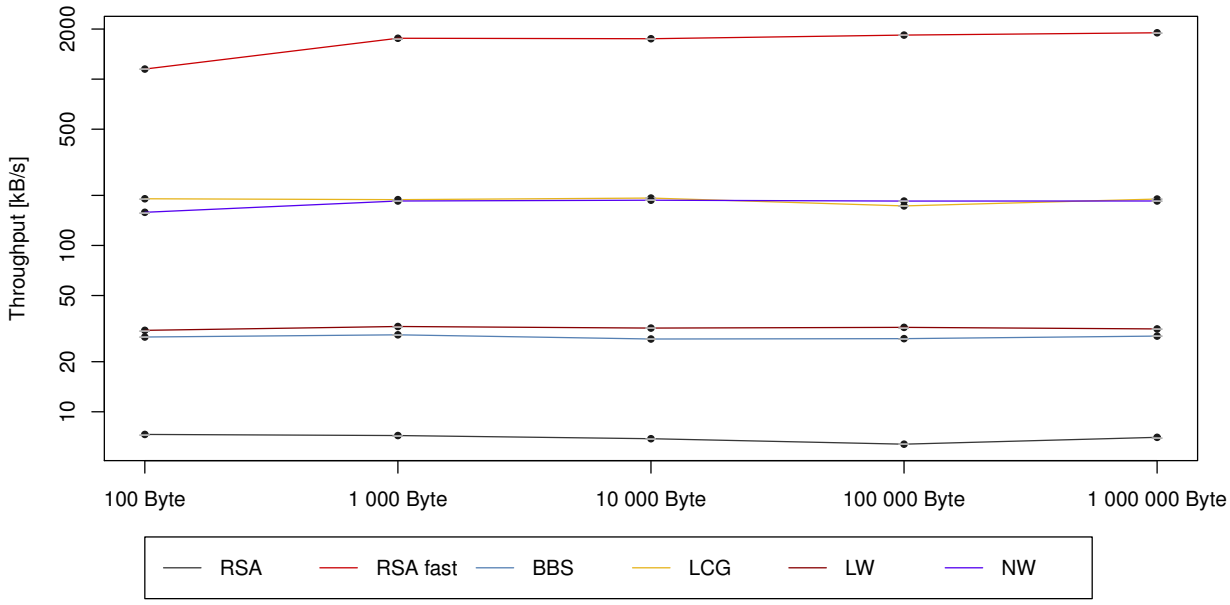

Figure 1: Generator throughput after initialization for different output lengths.

For benchmarking we used an odd s and for $z = (z_1, \ldots, z_s) \in \{0,1\}^s$, $f(z) =$ XOR of the bits of

$$(z_1, \ldots, z_{(s+1)/2})_2 + (z_{(s+1)/2}, \ldots, z_s)_2 \text{ in } \mathbb{Z}_{(s+1)/2},$$

where $(u)_2$ is the integer with binary representation u. This yields n pseudorandom bits in total. The computation can be done in parallel, since each bit only depends on the elements of S_i. In order to make the performance of Nisan-Wigderson PRG more stable and predictable we used only a single thread. The throughput of Nisan-Wigderson with $s = 131$, $k = 17161$ and $n = 2248091$, as in the table, does not account for the initialization time. Note that we used two different seeds for generating 512 kB of data, since a single seed only provides roughly 281 kB of data.

By pre-processing the design, time for generating the sets is not taken into account.

The generator is known to be pseudorandom if f is "hard" in a suitable sense. We do not claim this for our choice of f and thus not the pseudorandomness property.

4 Evaluation

To evaluate efficiency as well as statistical properties for the presented generators, we developed a framework that runs the software generators based on seed data from the PRG310-4. To this end, we implemented the generators in C, using the GMP library, see Granlund (2014), to accomplish large integer and floating point arithmetic. The evaluation framework sequentially runs all generators, reading from one true random source file of 512 kB and producing 512 kB each, while measuring the runtime of each generator and the byte entropy of each output.

	byte entropy [bit]	throughput w/o init [kB/s]	throughput w/ init [kB/s]	throughput w/o init normalized
PRG310-4, AES post-processing	7.99963	—	14.01806	1.94004
`/dev/random`, in the lab	7.99948	—	1.917×10^{-4}	2.653×10^{-5}
Littlewood	6.47244	33.66970	33.66602	4.61011
Linear congruential generator	7.99969	193.64313	79.84373	26.51392
Blum-Blum-Shub	7.99962	28.91291	28.37801	3.95880
RSA, fast	7.99966	1913.27484	1283.91594	261.96857
slow	7.99963	7.30345	7.22564	1
Nisan-Wigderson	7.99961	187.46153	116.23968	25.66753

Table 2: Overview of the results for warm-starting generators (w/o init). Cold-start (w/ init) throughput as comparison.

All algorithms were run on a Lenovo ThinkPad T530 with a Intel Core i7-3610QM @ 2.30GHz CPU with 12 GiB RAM. We used Linux Mint 17.1 Cinnamon 64-bit, Linux 3.13.0-55-generic, glibc 2.19, libgmp 5.1.3, libmpfr 3.1.2-p3 and gcc 4.8.4.

4.1 Isolating the generation process

In an attempt to increase comparability, we decided to split up the initialization and generation processes and measure the time for the generation only after a certain amount of data was generated. This way, the throughput of the generators had time to stabilize and we thus omit possible noise that is produced when the generator is first started, see the left part of Figure 1). To determine the appropriate amount of data to be generated before the measurement, we measured throughputs for increasing amounts of data so that we could see at which point the throughputs stabilize. To compare the two generation variants we have listed throughputs for both, along with byte entropy and normalized throughput of the additional measurements in Table 2. For parameter generation, only the fast variant of RSA actually generated a Blum integer. The variants RSA slow, as well as the Blum-Blum-Shub generator then used the computed value, explaining the considerable change for the timings of the slow variant of the RSA generator. Also for the linear congruential generator a prime modulus was generated. Note that Table 1 given in the abstract only lists the results for the generators *with* initialization.

4.2 Applying the NIST test suite

The NIST test suite, see Rukhin *et al.* (2010), is a collection of statistical tests, aimed at checking blocks of pseudorandomly generated data for statistical irregularities. The user specifies the tests to be run on the data, as well as the parameters of the tests and the block size used for processing the data. The output is an enumeration of the applied tests, the number of blocks for which each test succeeded and a p-value, providing a confidence level for the correctness of the test results. For each test, the NIST specification gives a minimum p-value that indicates whether the input data should be regarded as random.

For data to be considered statistically random, a minimal number of blocks must pass various

tests successfully. Each test has its own minimal block size. For instance, Maurer's universal test, see Maurer (1992), requires blocks to be roughly 300 kB at least.

With suitable parameters, all but the Littlewood generator did pass all tests. As the byte entropy of the outputs of the Littlewood generator already indicated statistical weaknesses, this does not come as a surprise. The fact that the cryptographically insecure linear congruential generator passes these tests with flying colors substantiates the well-known opinion that such tests are of little cryptographic significance.

5 Conclusion

We implemented a number of software random generators and compared their performance to physical generators. `/dev/random` is way too slow to be of practical use. The generator PRG310-4 is roughly as good as our Blum-Blum-Shub implementation. However, both are surpassed by the RSA generator when run with a fast parameter set, which offers the same level of security.

The most interesting — and surprising to us — result is that number-theoretic methods outperform hardware-based approaches by far. Their additional advantage is security under standard complexity assumptions such as the hardness of factoring certain integers. However, they still need random seeds. This suggests a profitable symbiosis of hardware-generated seeds and number-theoretic high throughput — rather the reverse of the situation in other cryptographic contexts, say, the Diffie-Hellman exchange of keys for fast AES encryption.

6 Future work

We decided on parameter choices that look reasonable to us, but many alternatives are possible. Do other parameter settings lead to qualitatively different results?

Practical use of our findings has not taken place yet. Depletion of `/dev/random` is a realistic issue — workarounds for implied problems even suggest using `/dev/urandom` as a physical generator, see Searle (2008). However, prohibiting the use of `/dev/urandom` for key generation is under debate, see Bernstein (2014). From the BSI's point of view, `/dev/urandom` fulfills all requirements for NTG.1 as defined in Killmann & Schindler (2011), except for NTG.1.1, see Müller *et al.* (2014).

Given benchmarking results and consumed entropy, using `/dev/random` for seed generation and then running a secure software PRG looks promising. Kernel based implementations of the algorithms we investigated are not available as of today. Benchmarking and field testing are yet to be done. As a next step, implementing and testing on kernel level using optimized implementations is recommended.

References

Werner Alexi, Benny Chor, Oded Goldreich & Claus P. Schnorr (1988). RSA and Rabin functions: Certain Parts are As Hard As the Whole. *SIAM Journal on Computing* **17**(2), 194–209. ISSN 0097-5397. URL `http://dx.doi.org/10.1137/0217013`.

Frank Bergmann (2014). Physikalische Zufallssignalgeneratoren für kryptografische Applikationen. URL `http://www.ibbergmann.de/`.

D. J. Bernstein (2014). cr.yp.to: 2014.02.05: Entropy Attacks! URL http://blog.cr.yp.to/20140205-entropy.html. Accessed: 27th June 2015.

L. Blum, M. Blum & M. Shub (1986). A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing* **15**(2), 364–383. URL http://dx.doi.org/10.1137/0215025.

Joan Boyar (1989). Inferring Sequences Produced by Pseudo-Random Number Generators. *Journal of the ACM* **36**(1), 129–141. URL http://dx.doi.org/10.1145/58562.59305.

Aleksei Burlakov, Johannes vom Dorp, Sarah Hillmann, Michael Link, Jan Lühr, Simon Schneider & Sven Zemanek (2015). Comparative analysis of pseudorandom generators: Source code. BitBucket. URL https://bitbucket.org/sirsimonrattle/15ss-taoc.

R. Fischlin & C. P. Schnorr (2000). Stronger Security Proofs for RSA and Rabin Bits. *Journal of Cryptology* **13**(2), 221–244. URL http://dx.doi.org/10.1007/s001459910008. Communicated by Oded Goldreich.

Torbjörn Granlund (2014). The GNU Multiple Precision Arithmetic Library. C library. URL https://gmplib.org/. Accessed: 25th June 2015.

Zvi Gutterman, Benny Pinkas & Tzachy Reinman (2006). Analysis of the Linux Random Number Generator. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, 371–385. IEEE Computer Society. ISBN 0-7695-2574-1. URL http://dx.doi.org/10.1109/SP.2006.5. Also available at http://eprint.iacr.org/2006/086.

Johan Hastad & Adi Shamir (1985). The Cryptographic Security of Truncated Linearly Related Variables. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, 356–362. ACM, New York, NY, USA. ISBN 0-89791-151-2. URL http://doi.acm.org/10.1145/22145.22184.

Wolfgang Killmann & Werner Schindler (2011). A proposal for: Functionality classes for random number generators. Anwendungshinweise und Interpretationen zum Schema AIS 20/AIS 31, Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany. URL https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile. Version 2.0.

Patrick Lacharme, Andea Röck, Vincent Strubel & Marion Videau (2012). The Linux Pseudorandom Number Generator Revisited. URL https://eprint.iacr.org/2012/251.pdf. Accessed: 27th June 2015.

D. H. Lehmer (1951). Mathematical methods in large-scale computing units. In *Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery, 13–16 September 1949*, volume 26 of *Annals of the Computation Laboratory of Harvard University*, 141–146. Harvard University Press, Cambridge, Massachusetts. URL https://archive.org/details/proceedings_of_a_second_symposium_on_large-scale_.

J. E. Littlewood (1953). *A Mathematician's Miscellany.* Methuen & Co. Ltd., London, 136 .

Ueli M. Maurer (1992). Protocols for Secret Key Agreement by Public Discussion Based on Common Information. In *Advances in Cryptology: Proceedings of CRYPTO 1992,* Santa Barbara, CA, Ernest F. Brickell, editor, number 740 in Lecture Notes in Computer Science, 461–470. Springer-Verlag. ISSN 0302-9743.

Stephan Müller, Gerald Krummeck & Mario Romsy (2014). *Dokumentation und Analyse des Linux-Pseudozufallszahlengenerators.* BSI. URL `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/LinuxRNG/LinuxRNG_Studie.pdf`. Accessed: 29th June 2015.

Noam Nisan & Avi Wigderson (1994). Hardness vs Randomness. *Journal of Computer and System Sciences* **49**, 149–167. Conference version in *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, White Plains, NY, 2–11. IEEE Computer Society Press.

Joan B. Plumstead (1982). Inferring a Sequence Generated by a Linear Congruence. *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science,* Chicago IL 153–159. URL `http://dx.doi.org/10.1109/SFCS.1982.73`.

Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray & San Vo (2010). *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.* U.S. Department of Commerce / National Institute of Standards and Technology. URL `http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf`. Special Publication.

Werner Schindler & Wolfgang Killmann (2003). Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications. In *Cryptographic Hardware and Embedded Systems — CHES 2002*, Jr. Burton S. Kaliski, Çetin K. Koç & Christof Paar, editors, volume 2523 of *LNCS*, 431–449. Springer-Verlag, Berlin, Heidelberg. ISSN 0302-9743 (Print), 1611-3349 (Online). URL `http://dx.doi.org/10.1007/3-540-36400-5_31`.

Chris Searle (2008). Increase entropy on a 2.6 kernel linux box. URL `https://www.chrissearle.org/2008/10/13/Increase_entropy_on_a_2_6_kernel_linux_box/`. Accessed: 27th June 2015.

Adi Shamir (1983). On the Generation of Cryptographically Strong Pseudorandom Sequences. *ACM Trans. Comput. Syst.* **1**(1), 38–44. ISSN 0734-2071. URL `http://doi.acm.org/10.1145/357353.357357`.

Damien Stehlé (2004). Breaking Littlewood's Cipher. *Cryptologia* **XXVIII**(4), 341–357. URL `http://dx.doi.org/10.1080/0161-110491892971`.

Ron Steinfeld, Josef Pieprzyk & Huaxiong Wang (2006). On the Provable Security of an Efficient RSA-Based Pseudorandom Generator. In *Advances in Cryptology: Proceedings of ASIACRYPT 2006,* Shanghai, China, Xuejia Lai & Kefei Chen, editors, volume 4284 of *Lecture Notes in Computer Science*, 194–209. Springer-Verlag, Berlin, Heidelberg. ISBN 978-3-540-49475-1. ISSN 0302-9743 (Print) 1611-3349 (Online). URL `http://dx.doi.org/10.1007/11935230_13`.

D. B. Wilson (1979). Littlewood's cipher. *Cryptologia* **3**, 120–121 and 172–176.

GliFreD: Glitch-Free Duplication
– Towards Power-Equalized Circuits on FPGAs –

Alexander Wild, Amir Moradi and Tim Güneysu

Horst Görtz Institute for IT-Security,
Ruhr-Universität Bochum

Hiding is known as common class for countermeasures to protect against Side-Channel Analysis (SCA). A subset of hiding countermeasures aims at equalizing the power consumption of the cryptographic device to keep it independent from the processed data – thwarting attacks such as Differential Power Analysis (DPA). These countermeasures, also known as DPA-resistant logic styles, usually implement the Dual-Rail Precharge (DRP) concept. Examples for this are SABL, WDDL, DRSL, MDPL, iMDP that are specifically tailored to be used in Application-Specific Integrated Circuit (ASIC) devices. However, due to predefined structures and restrictions in routing, the techniques of these schemes cannot be easily applied to Field Programmable Gate Arrays (FPGAs). Therefore, most of the efforts to equalize the power consumption on FPGAs have been put in the direction of *duplication*. Fortunately, an FPGA contains similar blocks formed by a couple of slices with (nearly) equal inter- and intraconnections. Hence, re-instantiating a part of a circuit at another location on the FPGA and converting it to its dual function seems to be a viable option. Previous works investigated this concept of duplication, but all reported schemes still show some vulnerabilities against certain power analysis attacks.

This work aims to design a scheme that rules out previous weaknesses to provide an SCA-resistant implementation of cryptographic circuts on FPGAs. Our scheme, denoted as *GliFreD*, avoids (1) glitches in combinatorial circuits, (2) forms a pipeline architecture, and (3) efficiently instantiates the duplication concept. We show in practical experiments on a Xilinx Spartan-6 FPGA how to combine Xilinx design tools and RapidSmith to finally convert an unprotected circuit into a corresponding DPA-protected one under the definitions of the GliFreD scheme.

Side-channel analysis of the converted circuits implemented on the SAKURA-G platform indicates the success of GliFreD to significantly mitigate the success of DPA attacks. We further elaborate the limitations of the duplication concept and provide reasons for the leakages that cannot be completely avoided.

Integral Attack on the ASASA Block Cipher Construction

Itai Dinur*, †Orr Dunkelman, Thorsten Kranz‡ and Gregor Leander‡

*École Normale Supérieure †University of Haifa ‡Ruhr University Bochum

We consider the problem of recovering the internal specification of a general SP-network consisting of three linear layers (A) interleaved with two Sbox layers (S) (denoted by ASASA for short), given only black-box access to the scheme. The decomposition of such general ASASA schemes was first considered at ASIACRYPT 2014 by Biryukov et al. [1] who used the alleged difficulty of this problem to propose several concrete block cipher designs as candidates for white-box cryptography.

We present and analyze an integral attack [2] on general ASASA schemes that significantly outperforms the analysis of Biryukov et al.

The attack starts with choosing random linear subspaces and then uses the integral property as a distinguisher to construct certain subspaces that enable us to decompose the ASASA construction.

As a result, we are able to break all the proposed concrete ASASA constructions with practical complexity. For example, we can decompose an ASASA structure that was supposed to provide 64-bit security in roughly 2^{28} steps, and break the scheme that supposedly provides 128-bit security in about 2^{41} time.

References

[1] Alex Biryukov and Charles Bouillaguet and Dmitry Khovratovich. Cryptographic Schemes Based on the ASASA Structure: Black-Box, White-Box, and Public-Key (Extended Abstract). *ASIACRYPT 2014, December 2014.*

[2] Itai Dinur and Orr Dunkelman and Thorsten Kranz and Gregor Leander. Decomposing the ASASA Block Cipher Construction. *Cryptology ePrint Archive, Report 2015/507.*

Higher-Order Polynomial Masking Hardware Implementations of AES

Fabrizio De Santis, Tobias Bauer and Georg Sigl

Technische Universität München
Munich, Germany
{desantis,tobias.bauer,sigl}@tum.de

Polynomial masking is a glitch-free higher-order masking scheme based upon the Shamir's secret sharing scheme and multi-party computation protocols to protect cryptographic implementations against side-channel analysis. Polynomial masking has been originally introduced at CHES 2011 [PR11], while a first prototype implementation of a *first-order* polynomial masking AES on FPGA appeared at CHES 2013 [MM13].

In this work, we take a step forward implementing more efficient *higher-order* polynomial masking implementations of AES in hardware. In particular, we provide the following contributions: first, we provide new addition chains for inversion in $\mathsf{GF}(2^8)$ and $\mathsf{GF}(2^4)$ which lead to faster and more compact circuit implementations. Then, we investigate various design trade-offs for higher-order polynomial masking AES implementations introducing a new serialized design over $\mathsf{GF}(2^4)$. Eventually, we provide synthesis and Electro-Magnetic (EM) field side-channel analysis results.

References

[PR11] Emmanuel Prouff and Thomas Roche. Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. *Cryptographic Hardware and Embedded Systems – (CHES 2011)*, LNCS Springer vol. 6917, 63–78, 2011.

[MM13] Amir Moradi and Oliver Mischke. On the Simplicity of Converting Leakages from Multivariate to Univariate. *Cryptographic Hardware and Embedded Systems – (CHES 2013)*, LNCS Springer vol. 8086, 1–20, 2013.

On the Design of Stream Ciphers with Shorter Internal States

Vasily Mikhalev and Frederik Armknecht

University of Mannheim
Germany

In order to be resistant against certain time-memory-data-tradeoff (TMDTO) attacks [3], the internal state size of a stream cipher should be at least twice the security parameter. As memory gates are usually the most area and power consuming components, this implies a sever limitation with respect to possible lightweight implementations.

We recently revisited this rule at FSE 2015 [1], and suggested a new design approach which enables stream ciphers with shorter internal states. To prove the concept, a new stream cipher named Sprout was developed[1]. It received a serious attention in the crypto community and several weaknesses were indicated [7, 6, 4, 5, 2]. Although these weaknesses allow for efficient attacks against Sprout, none of the papers appeared so far dispute the correctness of the main concept for the stream cipher design that was suggested, meaning that the secure variants are probably possible.

In this work we analyze the discovered attacks and systematize the weaknesses found in Sprout. Then we suggest the possible countermeasures against the attacks and analyze how hardware friendly they are. Finally we try to combine these countermeasures into one design of a new secure stream cipher with shorter internal state.

References

[1] Frederik Armknecht and Vasily Mikhalev. On lightweight stream ciphers with shorter internal states, In *Fast Software Encryption FSE* , 2015

[2] Subhadeep Banik. Some results on Sprout, Technical report,Cryptology ePrint Archive, Report 2015/327, 2015. *http://eprint. iacr. org*

[3] Alex Biryukov and Adi Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers *In Advances in Cryptology ASIACRYPT 2000,*pages 113. Springer, 2000.

[4] Muhammed Esgin and Orhun Kara Practical Cryptanalysis of Full Sprout with TMD Tradeoff Attacks, Technical report,Cryptology ePrint Archive, Report 2015/289, 2015. *http://eprint. iacr. org*

[5] Yonglin Hao. A related-key chosen-iv distinguishing attack on full sprout stream cipher Technical report,Cryptology ePrint Archive, Report 2015/231, 2015. *http://eprint. iacr. org*

[6] Virginie Lallemand and Maria Naya-Plasencia Cryptanalysis of full Sprout To appear In *Advances in CryptologyCRYPTO 2015*

[7] Subhamoy Maitra and Santanu Sarkar and Anubhab Baksi and Pramit Dey. Key recovery from state information of sprout: Application to cryptanalysis and fault attack Technical report,Cryptology ePrint Archive, Report 2015/236, 2015. *http://eprint. iacr. org*

Red/Green Hardware-Security Technologies for Internet-of-Things (IoT): ARM TrustZone and Security Controller

Christian Lesjak*, Daniel Hein† and Johannes Winter†

* Design Center Graz
Infineon Technologies Austria AG
Graz, Austria

† Inst. of Applied Information Processing and Communications
Graz University of Technology
Graz, Austria

Dual execution environments harness the security by isolation paradigm by offering a secured environment which provides authenticity, integrity and privacy for executing sensitive processes. We refer to this secured execution environment as the green world, which is isolated from the red world. The red world denotes the normal or rich (in terms of general purpose processing capability) execution environment. Among various software-based approaches, ARM TrustZone and Security Controller are two hardware-security based technologies for security by isolation. The ARM TrustZone security extensions, marketed as ARM TrustZone, provide a second virtual processor backed by hardware access controls to logically separate the red and green world. The Security Controller as a dedicated integrated circuit provides a secured execution environment in a hardware security element with dedicated processor and memory.

We evaluate and compare ARM TrustZone and the Security Controller in regard to their provided security, flexibility, and performance. Both technologies have recently gained attention in the industrial research community: Fitzek *et al.* [FW+15] have presented the ANDIX research OS for ARM TrustZone, while Lesjak *et al.* [LH+15] have used a Security Controller to protect the Transport Layer Security (TLS) client authentication step in industrial equipment. Thus, for our comparison, we design and implement security components for an industrial authentication scenario, and illustrate how to partition these components within both, a TrustZone and a Security Controller based system.

Our results indicate that the ARM TrustZone based approach promises greater flexibility and performance, but only the Security Controller strongly protects against physical attacks. We argue that the best technology actually depends on the use case and propose a hybrid approach that maximizes the security for our our exemplary industrial use case. We believe that the insights we gained will help introducing advanced security mechanisms into future IoT applications and industrial systems.

References

[LH+15] C. Lesjak, D. Hein, M. Hofmann, M. Maritsch, A. Aldrian, P. Priller, T. Ebner, T. Ruprechter, and G. Pregartner. Securing smart maintenance services: hardware-security and TLS for MQTT. *IEEE INDIN 2015*.

[FW+15] A. G. Fitzek, J. Winter, F. Achleitner, and D. Hein. The ANDIX research OS - ARM TrustZone meets industrial control systems security. *IEEE INDIN*, 2015.

Self-Secure Exponentiation Countermeasures Against Fault Attacks and Power Analysis for RSA-CRT[1]

Ágnes Kiss* and Juliane Krämer†◇

*ENCRYPTO, †CDC ◇SecT
TU Darmstadt TU Berlin
Darmstadt Berlin

While performing fault analysis, an adversary induces computational errors into the algorithm and gains information about the secret key from the result. In [BDL97], Bellcore researchers proved that fault analysis was a powerful method to break cryptographic implementations. RSA, the most widely used public-key cryptosystem was subject to their fault attack, the Bellcore attack, especially when implemented using the Chinese remainder theorem (RSA-CRT). Therefore, it became essential to find countermeasures that prevent fault analyses. Two major techniques appeared in the last decades, one of which uses so-called *self-secure exponentiation algorithms*.

We classify all existing RSA-CRT countermeasures against the Bellcore attack that use binary self-secure exponentiation algorithms. We test their security against a generic adversary by simulating fault injections at all possible fault locations. Using our half-automated testing method, we find that most of the countermeasures do not provide sufficient security against fault attacks, but are insecure in a generic fault model. Besides fault attacks, the countermeasures consider the vulnerability of the exponentiation algorithms against power analysis and safe-error attacks as well. We investigate how additional measures can be included to counter all possible fault injections and power analyses, without introducing new vulnerabilities into the countermeasures.

There are three exponentiation algorithms used to construct self-secure exponentiation countermeasures: there exist techniques that use the *Montgomery ladder* [G06, FV06], algorithms that rely on the *square-and-multiply-always exponentiation* [BNP07, BHT09] and methods that make use of *double exponentiation* [R09, LRT14]. In this talk, we summarize our classification and testing results for all the three categories and present an improvement on one of the countermeasures.

References

[BDL97] Boneh, Dan and DeMillo, Richard A and Lipton, Richard J. On the importance of checking cryptographic protocols for faults. *Advances in Cryptology – Eurocrypt'97*, Springer, 37–51, 1997.

[BHT09] Boscher, Arnaud and Handschuh, Helena and Trichina, Elena. Blinded Fault Resistant Exponentiation Revisited. *Fault Diagnosis and Tolerance in Cryptography, FDTC 2009*, IEEE, 2009.

[BNP07] Boscher, Arnaud and Naciri, Robert and Prouff, Emmanuel. CRT RSA Algorithm Protected Against Fault Attacks. *Information Security Theory and Practices*, Springer, 229–243, 2007.

[FV06] Fumaroli, Guillaume and Vigilant, David. Blinded Fault Resistant Exponentiation. *Fault Diagnosis and Tolerance in Cryptography, FDTC 2006*, Springer , 62–70, 2006.

[G06] Giraud, Christophe. An RSA Implementation Resistant to Fault Attacks and to Simple Power Analysis. *IEEE Trans. Computers*, vol. 55/9, 1116–1120, 2006.

[LRT14] Le, Duc-Phong and Rivain, Matthieu and Tan, Chik How. On Double Exponentiation for Securing RSA against Fault Analysis. *Topics in Cryptology, CT-RSA 2014*, Springer, 152–168, 2014.

[R09] Rivain, Matthieu. Securing RSA against Fault Analysis by Double Addition Chain Exponentiation. *Topics in Cryptology, CT-RSA 2009*, Springer, 459–480, 2009.

[1]This work was partially supported by the European Union's Seventh Framework Program (FP7/2007-2013) grant agreement n. 609611 (PRACTICE).

Ascon – Submission to the CAESAR Competition

Christoph Dobraunig[1], Maria Eichlseder[1], Florian Mendel[1], Martin Schläffer[1,2]

[1]IAIK, Graz University of Technology, Austria
[2]Infineon Technologies AG, Austria

Ascon [1] is an authenticated encryption scheme submitted to the ongoing CAESAR competition [2]. The goal of CAESAR is to select a portfolio of authenticated encryption schemes from more than 50 submissions which are suitable for widespread adoption. The main design goal and trade-off for Ascon is to have a very low memory footprint in both hardware and software, while still being fast and providing high confidence in its security. Ascon offers security proofs, bounds for the resistance against a large class of attacks and is easy to analyze. Additionally, Ascon was designed to simplify efficient implementations of side-channel countermeasures.

Ascon is based on the sponge construction [3] using a MonkeyDuplex [4] like mode of operation. The permutation of Ascon uses an iterated substitution-permutation-network (SPN), which provides good cryptographic properties and fast diffusion at a low cost. To provide these properties, the main components of Ascon are inspired from standardized and well analyzed primitives. The substitution layer uses an improved version of the S-box used in the χ mapping of Keccak [5]. The permutation layer uses a linear functions similar to the Σ functions of SHA-2.

The security of the Ascon mode of operation has been proven in [6]. Additionally, Ascon has bounds for its security and provides a rich set of security analysis [7]. Amongst others, cube-like, differential and linear cryptanalysis techniques have been used to evaluate the security of Ascon. In this talk we present Ascon, its design ideas and trade-offs, and discuss the most recent implementation and cryptanalysis results.

References

[1] Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon. Submission to the CAESAR competition: `http://ascon.iaik.tugraz.at` (2014)

[2] The CAESAR committee: CAESAR: Competition for authenticated encryption: Security, applicability, and robustness (2014), `http://competitions.cr.yp.to/caesar.html`

[3] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Sponge Functions. ECRYPT Hash Workshop 2007, May 2007.

[4] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *LNCS*, pages 320–337. Springer, 2011.

[5] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Keccak specifications. Submission to NIST (Round 3), 2011.

[6] Jovanovic, P., Luykx, A., Mennink, B.: Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. Cryptology ePrint Archive, Report 2014/373 (2014), `http://eprint.iacr.org/2014/373`

[7] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of Ascon. In Kaisa Nyberg, editor, *Topics in Cryptology - CT-RSA 2015*, volume 9048 of *LNCS*, pages 371–387. Springer, 2015.

Side-Channel Protection with Dynamic Logic Reconfiguration and Randomized Look-Up Tables on FPGAs

Pascal Sasdrich

Horst Görtz Institute for IT-Security,
Ruhr-Unversität Bochum,
Bochum, Germany

Reconfigurability is a unique feature of modern FPGA devices to load hardware circuits just on demand. This also implies that a completely different set of circuits might operate at the exact same location of the FPGA at different time slots, making it difficult for an external observer or attacker to predict what will happen at what time.

In this work we present and evaluate a novel hardware implementation of the lightweight cipher PRESENT with built-in side-channel countermeasures based on dynamic logic reconfiguration. In our design we make use of Configurable Look-Up Tables (a special operation mode of common LUTs) integrated in modern Xilinx FPGAs to nearly instantaneously change hardware internals of our cipher implementation for improved resistance against side-channel attacks.

In a second approach, we extend this idea for AES by improving the Block Memory Content Scrambling (BMS), presented at CHES 2011. This scheme enables an effective way of first-order side-channel protection for cryptographic primitives at the cost of a significant reconfiguration time for the mask update. We now analyze alternative ways to implement dynamic first-order masking of AES with randomized lookup tables that can reduce this mask update time. The memory primitives we consider in this work include three distributed RAM components (again a special operation modes of common LUTs) and one BRAM primitive (RAMB8BWER). We provide a detailed study of the area and time overheads of each implementation technique with respect to the operation (encryption) as well as reconfiguration (mask update) phase.

We further compare the achieved security of each approache to prevent first-order side-channel leakages. Our evaluation is based on a state-of-the-art leakage assessment methodology known as t-test.

References

[1] P. Sasdrich, A. Moradi, O. Mischke, T. Gneysu. Achieving Side-Channel Protection with Dynamic Logic Reconfiguration on Modern FPGAs. In HOST 2015, pages 130-136, IEEE Computer Society, 2015

[2] P. Sasdrich, O. Mischke, A. Moradi, T. Gneysu. Side-Channel Protection by Randomizing Look-Up Tables on Reconfigurable Hardware. To appear in the proceedings of COSADE 2015.

Towards Miniaturized System-in-Package Contactless and Passive Authentication Devices featuring NFC

Norbert Druml, Juergen Schilling, Walther Pachler, Bernhard Roitner
Thomas Ruprechter, Holger Bock, and Gerhard Holweg

Design Center Graz
Infineon Technologies Austria AG
Graz, Austria

Today, the RFID/NFC technology is widely spread and applications can be found in our everyday life, for example, in the fields of payment, loyalty and coupons, transportation, healthcare, and access control (cf. [Fin02]). Furthermore, recent smart phones are equipped with NFC functionality in order to communicate with these RFID/NFC-enhanced tags. However, state-of-the-art contactless and passive authentication solutions implement relatively large coils outside of the chip. The minimum size is in the order of a few square centimeters, which limits their use for tagging of small-sized goods.

The present research undertaking aims to introduce a miniaturized contactless and passive authentication solution. This is achieved by integrating Infineon Technology's CIPURSE^{TM} Move chip, which is a state-of-the-art authentication solution featuring an open security standard, into embedded Wafer Level Ball Grid Array (eWLB) packages, cf. [BM+08], together with HF-antennas, ferrites, as well as discrete elements that improve HF-coupling characteristics. Thus, a System-in-Package authentication solution is given, cf. [PB+10].

Compared to state-of-art, our solution will provide better HF-coupling characteristics than Coil-on-Chip approaches, which will also enable a verification of authenticity of tagged products through NFC-enabled smart phones. Thanks to the miniaturized package sizes of 3x3 mm and 5x5 mm, the integration into various types of small products is enabled, such as jewelry, casings, consumable materials, etc. Furthermore, the integration of ferrites enables a deployment into metallic environments. Therefore, this miniaturized contactless authentication solution will open up whole new fields of applications.

References

[Fin02] K. Finkenzeller. RFID-Handbuch. *Hanser Verlag*, 2002.

[BM+08] M. Brunnbauer, T. Meyer, G. Ofner, K. Mueller, and R. Hagen. Embedded Wafer Level Ball Grid Array (eWLB). *33rd IEEE/CPMT International Electronic Manufacturing Technology Symposium (IEMT)*, pp.994-998, November 2008.

[PB+10] K. Pressel, G. Beer, T. Meyer, M. Wojnowski, M. Fink, G. Ofner and B. Roemer. Embedded wafer level ball grid array (eWLB) technology for system integration. *IEEE CPMT Symposium Japan*, August 2010.

Side-Channel Implications of Deterministic DSA-Signature Variants

Hermann Seuschek[†], Johann Heyszl[*], and Fabrizio De Santis[†]

[†] Technische Universität München
Munich, Germany
{hermann.seuschek,desantis}@tum.de

[*] Fraunhofer Institute AISEC
Munich, Germany
johann.heyszl@aisec.fraunhofer.de

Two recent proposals [1][2] suggest the use of so-called *deterministic* signatures for DSA and its elliptic curve-based variants. The core idea of *deterministic* signatures is to derive the required ephemeral value for the signature in a deterministic manner instead of using random numbers. The ephemeral value is derived from the message to be signed and the long-term secret signature key using an HMAC construction. This prevents possible vulnerabilities from low quality random numbers in actual implementations which is particularly important for small embedded devices in internet of things applications where the generation of high quality random numbers is difficult to achieve. Additionally, recent discoveries have raised partial skepticism, whether certain standardized methods for random number generation may have backdoors to weaken cryptographic operations, which can be eliminated in this way.

While we support the aim of the proposed determinism, we are concerned about the fact, that this has a significant influence on the implementation security. Due to the fact that the *long-term secret key is processed by a cryptographic hash function*, differential side-channel attacks can be mounted on this key with a much higher probability of success than in the previous case. This is because hash functions are far more difficult to protect against such attacks than the previous single point of secret key usage in the signature scheme, which is the linear integer arithmetic function to generate the second signature parameter. Countermeasures to protect hash functions against differential side-channel attacks would again need random numbers which is contrary to the proposals original intent. As a consequence and in the context of implementations which are prone to side-channel attacks, the need for random numbers cannot be fully removed with deterministic signatures, although the quality requirements of random numbers are lower because they are only needed in side-channel protection.

References

[1] D. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang: High-speed high-security signatures Journal of Cryptographic Engineering, Volume 2, Issue 2, pp 77-89, September 2012.

[2] T. Pornin: Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) Request for Comments 6979, Internet Engineering Task Force, August 2013.

Implementing Cryptographic Pairings on Infineon SLE 78

Peter Günther [1]

University of Paderborn

Bilinear pairings have become an important tool in cryptography. Today numerous schemes such as identity based encryption (IBE), attribute based encryption, or signatures with additional properties use pairings as their main building blocks. Many pairing based schemes are very well suited to embedded applications. For example with IBE, the expensive public key infrastructure of large scale systems like the internet of things can be significantly simplified. Hence, efficient implementations of pairings on embedded and resource constrained devices will become important in the future. Furthermore, in many pairing based schemes the secret key is one argument of the pairing. To protect this secret in an adversarial environment, implementations on smart cards are the standard solution. Cryptographic pairings are defined on subgroups of elliptic curves over finite fields $\mathbb{F}_q$. For primitives from elliptic curve cryptography (ECC) (*e.g.* the double and add algorithm), efficient implementations on smart cards exist. As we will explain in the talk, efficient implementations of pairings require more memory and more arithmetic operations than those primitives. This raises the question whether existing constrained embedded platforms such as smart cards are able to compute pairings. Furthermore, bottlenecks of current architectures have to be identified.

The first results in this direction show that it is indeed possible to compute pairings on existing smart card controllers. These examples cover different architectures from Philips/NXP [1], STMicroelectronics [2], or Atmel [3]. We implemented the eta pairing over binary fields on an Infineon SLE 78 controller. Our implementation is highly optimized for the cryptographic coprocessor of the device. This allows us to compute the eta pairing in 60 milliseconds for fields of size 1000 bits, in 100 milliseconds for fields of size 1500 bits, and in 160 milliseconds for fields of size 2000 bits. Our results show that pairings can efficiently be computed on current smart cards, but only if the pairing is selected carefully according to the available resources.

In this talk, we present our implementation and we also point to some bottlenecks of current smart card controllers with respect to pairing computations.

References

[1] Michael Scott, Neil Costigan, and Wesam Abdulwahab. Implementing cryptographic pairings on smartcards. In *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 134–147. Springer, 2006.

[2] Guido Bertoni, Luca Breveglieri, Liqun Chen, Pasqualina Fragneto, Keith A. Harrison, and Gerardo Pelosi. A pairing SW implementation for smart-cards. *Journal of Systems and Software*, 81(7):1240–1247, 2008.

[3] Leonardo B. Oliveira, Diego F. Aranha, Conrado Porto Lopes Gouvêa, Michael Scott, Danilo F. Câmara, Julio López, and Ricardo Dahab. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. *Computer Communications*, 34(3):485–493, 2011.

[1] This work was supported by the German Federal Ministry of Education and Research, grant 01 IS 10030 C.

Bounding the Differential Probability of SIMON

Christof Beierle
Horst Görtz Institute for IT-Security
Ruhr-Universität Bochum, Germany

In this work, we analyze the differential probability of SIMON-like functions. Considering the SIMON family of block ciphers [BSS+13] as a special case, we provide a proof on the resistance against differential attacks by upper bounding the probability of a differential characteristic to 2^{-2T+2} where T denotes the number of rounds. This is done by using some observations described in [KLT15]. Precisely, we use the fact that the algebraic degree of the non-linear part of the round function equals two such that the set of possible output differences defines an affine subspace depending on the input difference. This allows for considering just the Hamming Weights of the input differences. Interestingly, if $2n$ denotes the block length, this result is sufficient in order to bound the probability to 2^{-2n} for all full-round variants of SIMON. Thus, it guarantees security in a sense that one needs to have encryptions of more than the full codebook to find a specific differential characteristic.

Although there are much better bounds known, especiallly for a high number of rounds, they are still based on experimental search like using SAT/SMT solvers [KLT15]. This work is a step towards a more formal way of arguing on the resistance against differential attacks for SIMON-like designs.

The proof of the main result is based on follwoing fact, which is an implication of a Theorem described in [KLT15].

Lemma 1 *Let $\boldsymbol{\alpha}$ be an input difference into the SIMON-f function f_S of the Feistel construction. Then, for the differencial probability over f_S it holds that*

(1) if $\mathrm{wt}(\boldsymbol{\alpha}) = 1$, *then* $p_\alpha \leq 2^{-2}$ *(2) if* $\mathrm{wt}(\boldsymbol{\alpha}) \in \{2, 3\}$, *then* $p_\alpha \leq 2^{-3}$

(3) if $\mathrm{wt}(\boldsymbol{\alpha}) \geq 4$, *then* $p_\alpha \leq 2^{-4}$

It is thus sufficient to concentrate on low Hamming Weights upto three. One is now able to obtain the desired upper bound by checking several cases.

References

[KLT15] Stefan Kölbl and Gregor Leander and Tyge Tiessen. Observations on the SIMON block cipher family. *Advances in Cryptology–CRYPTO 2015 (to appear)*, Springer, 2015.

[BSS+13] Ray Beaulieu and Douglas Shors and Jason Smith and Stefan Treatman-Clark and Bryan Weeks and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. *IACR Cryptology ePrint Archive*, 2013/404, 2013.

Faster Secure Computation through Automatic Parallelization

Niklas Buescher and Stefan Katzenbeisser

Technische Universität Darmstadt

In the thirty years since Yao's seminal papers [1, 2], Secure Multiparty Computation (MPC) and Secure Two-Party Computation (TPC) have transitioned from purely theoretic constructions to practical tools. Even so, TPC based on Yao's garbled circuits has seen a lot of progress over the past decade, compared with generic computation, TPC is still multiple orders of magnitude slower. Observing the ongoing trend towards parallel hardware, e.g., many-core architectures on a single chip, in this work we investigate whether parallelism within Yao's protocol targeting security against semi-honest adversaries can be exploited to further enhance its performance. Therefore, we propose a practical parallelization scheme. Its advances over existing parallelization approaches are twofold.

First, we present a compiler that detects parallelism at the source code level and automatically transforms C code into parallel circuits. Namely, we extend CBMC-GC [3] - an ANIS-C compiler for STC - by a new source-to-source front-end, which decomposes parallel and sequential code regions. The then compiled circuits allow an efficient and scalable execution on parallel hardware.

Second, we present an extension to Yao's protocol to balance the computation costs of both parties. In the original protocol, using the defacto standard point-and-permute optimization [4, 5], the garbling party has to perform four times the cryptographic work than the evaluating party. Given the identified parallelism, the idea of our protocol is to divide the work in a symmetric manner between both parties by switching the roles of the garbling and evaluating party. This *inter-party parallelization (IPP)* approach leads to significant efficiency increases already on single-core hardware without compromising security.

Multiple implementations illustrate the practicality of our approach. For example, we report a speed-up of 4.36 on 4 cores for the example application of modular exponentiation. Moreover, we show how IPP can be used to exploit bi-directional communication in bandwidth limited environments to further reduce the runtime of Yao's protocol by more than 30%.

References

[1] Yao, A. C. Protocols for secure computations. In *Symposium on Foundations of Computer Science SFCS* (1982).

[2] Yao, A. C. How to generate and exchange secrets. In *Symposium on Foundations of Computer Science SFCS* (1986).

[3] Holzer, A., Franz, M., Katzenbeisser, S., and Veith, H. Secure Two-Party Computations in ANSI C. In *ACM Conference on Computer and Communications Security CCS* (2012).

[4] Beaver, D., Micali, S., and Rogaway, P. The Round Complexity of Secure Protocols. In *ACM Symposium on Theory of Computing STOC* (1990).

[5] Malkhi, D., Nisan, N., Pinkas, B., and Sella, Y. Fairplay-Secure Two-Party Computation System. In *USENIX Security Symposium* (2004).

23rd Crypto-Day

ESCRYPT GmbH

December 10 and 11, 2015

Berlin, Germany

On December 10 and 11, 2015, the interest group "Angewandte Kryptographie" of Gesellschaft für Informatik e. V. will host the twenty-second *Crypto-Day*.

Ambition and Program: The Crypto-Day aims at providing an opportunity for early-stage researchers in the field of cryptography and IT-security to exchange knowledge and establish networks to universities as well as to industry (e.g. for collaboration across Germany, or to find out about research internships and post-doc positions). Therefore, we invite students, doctoral candidates, and experienced researchers to present their research results or research ideas in the form of 20 minute presentations on this upcoming Crypto-Day.
The ESCRYPT GmbH will host the event and provide insights into real world challenges in embedded security – a discipline at the confluence of cryptography, electrical engineering, and computer science.

Topics: The presented talks shall cover a broad spectrum from the field of cryptography or IT-security. We invite presentations of work-in-progress, contributions, which may be submitted to a conference, or summarize findings from a thesis or dissertation.
Submitted articles corresponding to the presentations will be arranged in a technical report. Therefore, submissions will be quotable publications and will be published on the web page. Observe that this does not forbid the publication of the result at other conferences or journals.

Attendance: There are **no participation fees**.

Submission: Please submit an abstract of your talk (**one DIN A4 page**). To simplify generation of the technical report, we request you to only use the LaTeX template of the cryptography group and to provide the PDF file additionally to the LaTeX sources.

Further information related to the venue, as well as to the registration and submission process will be provided timely on the web page.

Further Information (Program, Venue, LaTeX-template): http://www.kryptotag.de

Submission:	Until **November 23, 2015** per email
Registration:	Until **November 23, 2015** per email
Organisation:	Frederik Armknecht, Universität Mannheim Moritz Minzlaff, ESCRYPT GmbH
Contacts:	armknecht@uni-mannheim.de moritz.minzlaff@escrypt.com