FP7-ICT-SEC-2009-5
Contract no.: 258754
www.TAMPRES.eu

TAMPRES

Deliverable D3.2

Lightweight cryptographic technologies and their security

Editor:	Matt Robshaw
Deliverable nature:	Deliverable, Report (R)
Dissemination level: (Confidentiality)	Public
Contractual delivery date:	31 December 2011
Actual delivery date:	3 October 2011
Suggested readers:	Consortium
Version:	1.0
Total number of pages:	21
Keywords:	Lightweight cryptography.

Abstract

We provide an overview of cryptographic solutions likely to be relevant to sensor node deployment, including a particular focus on lightweight cryptography. Within the deliverable, we identify the technologies that will be considered for ongoing study and deployment within project TAMPRES.

Disclaimer

Impressum

TAMPREs - Tamper Resistant Sensor Node

TAMPRES

WP3 “Side Channel and Fault Attack Resistance”

D3.2 - Lightweight cryptographic technologies and their security

[Editor: Name, company] Matt Robshaw, Applied Cryptography Group, Orange Labs, France

List of authors

Company	Author
Orange Labs	Matt Robshaw
Orange Labs	Tiphaine Romand
UCL	Marcel Medwed
UCL	Francois Koeune
UCL	Xavier-Francois Standaert
NXP	Ventzi Nikov

Contents

1 Introduction

In this document we consider the suitability of different cryptographic algorithms for constrained environments. In particular we consider sensor networks where the nodes of said network are likely to be somewhat limited in power consumption or computational power and this may impact the security that is implemented.

The field of cryptographic design for constrained devices has evolved massively in recent years and it is sometimes described in the literature as *low-cost* or *lightweight* cryptography. While it may be tempting to expect such designs to also deliver reduced security, this need not necessarily be the case. Instead, these new primitives have been designed from scratch for efficient hardware implementation rather than attempting to provide a broad performance profile on a range of implementation platforms.

Perhaps the biggest driver for lightweight cryptography, at least in the literature, has been the development of RFID-tag based applications. However many of the same algoritms that have been proposed there can be considered for sensor networks. That said, we typically have a bit more freedom in the algorithms we might choose for sensor nodes—since the devices are more sophisticated than a basic RFID tag—and we can also consider some of the more efficient standardised algorithms as a deployment option.

It is often stated that adding security to constrained devices is difficult because the necessary low costs of a device limits the amount of silicon available. While this captures part of the problem, the difficulties are a far more complex mix of factors. An important issue is the availability of power, or the lack of it. In passive systems or ones that are battery-powered, we would like to use algorithms that are energy-efficient. And while there can be a range of area/energy trade-offs for an implementation, the more area-efficient implementations tend to be "serial" rather than "parallel" which, in turn, carry a time penalty. Depending on the application this can be an important consideration.

In this report we make a brief survey of the range of cryptographic algorithms that might be of interest to project TAMPRES. We make our recommendations for the technologies that might be interesting to explore further and, a key feature of project TAMPRES, we provide pointers towards a significant item of work; that of designing and testing efficient counter-measures to a range of side-channel cryptanalysis.

2 Security Goals

The application and threat environment have an overwhelming influence on the security goals we are interested in. What might be appropriate for one application can be entirely pointless for another. Even worse, some solutions for one goal can create a problem for another.

In general terms, we might be interested in the following security goals:

- data confidentiality: typically provided by *encryption*,
- data authentication: typically provided by a *message authentication code* when using symmetric cryptography or *digital signatures* in the asymmetric case,
- unilateral or mutual entity authentication: typically provided by a *challenge-response* or a *commitment-challenge-response protocol* that can be based on symmetric or asymmetric cryptography.

In project TAMPRES our goal is to provide the most flexible solutions for the widest range of applications. For work at such a low level, *i.e.* in the choice of cryptographic algorithm, we prefer to avoid future restrictions on our work and we will identify, and work with, the most versatile families of algorithms for our work within TAMPRES.

3 Algorithms and Protocols

At the heart of many security solutions lies a cryptographic primitive. There are many different types and not all would be suitable for efficient implementation. Generally speaking, cryptographic algorithms can be organised according to the way they use key material. Some algorithms require all the participants (the sender and the receiver for instance) to share the same secret key; these are referred to as *secret key* or *symmetric* algorithms. Other algorithms allow the participants to avoid sharing secret key material between all participants; these are referred to as *public key* or *asymmetric* algorithms.

3.1 Symmetric algorithms.

Participants share the same secret key. The relative performance of different symmetric algorithms is presented in Tables 1 and 2.

BLOCK CIPHER. This primitive is arguably the most important tool for the cryptographer. Not only can a block cipher be used directly for encryption, but it can be used for tag authentication within a *challenge-response* protocol, see Section 3.3. A block cipher can also, when used in an appropriate way, be used to construct all the other symmetric primitives. One of the most promising recent developments is the block cipher PRESENT [11] which was specifically designed for constrained hardware. However research continues and new proposals are frequently made [14].

STREAM CIPHER. These primitives have the reputation for being smaller and lighter than block ciphers, but with developments in new block cipher designs [11] it is not clear that such a distinction can still be made. We can use a block cipher to build a stream cipher [58] and this may well be the preference. However there are dedicated designs with two of the most promising being the Grain family [35] and Trivium [13]. These appear in the eSTREAM portfolio [75], along with Mickey v2 [7], and they provide interesting opportunities for efficient hardware implementation. Finally we mention the proposal QUAD [6] which is interesting for its marriage of provable security and the potential for reasonably low-area implementations.

HASH FUNCTION. The design of a secure hash function is a significant cryptographic problem [59]. For constrained devices it is even more challenging. Hash functions compress arbitrary-length inputs to a fixed-length (short) output while satisfying certain security properties [52]. Dedicated designs tend to be too large for tags and while hash functions built from a compact block cipher are smaller [11], they offer potentially unacceptable levels of security. Recently some promising new designs have been proposed [3] but they have not really received sufficient scrutiny. Ironically, theoreticians sometimes call upon hash functions in the design of privacy-preserving protocols targetted at RFID deployment; however the practicality of instantiating such protocols is rarely clear.

MESSAGE AUTHENTICATION CODE. A cryptographic checksum has a wide variety of authentication applications. Often MACs are built out of other primitives [52, 56, 57] and it is most likely that we would build a MAC out of a low-cost block cipher. However there are also dedicated designs. While many of these are proprietary, the proposal SQUASH was intended for RFID tags [79]. It is unclear, though, how to choose good parameter sets to balance performance and security [68].

3.2 Asymmetric algorithms.

The participants in a cryptographic exchange, say the sender and receiver, do not use the same key material, some of which can be made public.

ENCRYPTION. Depending on the application there might be calls to add public-key encryption to sensor networks; for instance to exchange secret shared keys or to provide a way to return data to a central node/server.

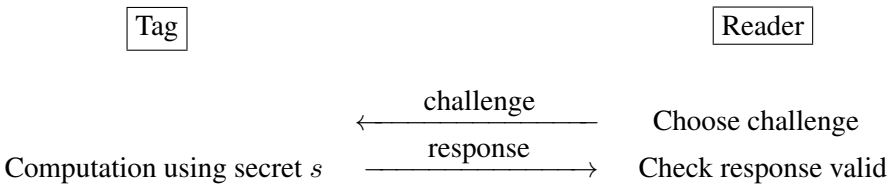
DIGITAL SIGNATURES. The digital signature can be used to demonstrate the authenticity of a device if used in a *challenge-response* protocol. The most promising candidate for dynamic signatures on constrained devices would probably be ECDSA [60], the elliptic-curve variant of the DSA signature scheme. However, as shown in Section 4.2, the physical costs remain large. Application considerations might also suggest a digital signature for data authentication.

IDENTIFICATION SCHEMES. Such schemes allow a tag to "prove" that it contains a tag-specific secret during an interactive *commitment-challenge-response* protocol with the reader, see Section 3.3. These schemes can be converted to give digital signatures [52], though such a conversion is rarely done in practice and identification schemes are deployed on their own terms.

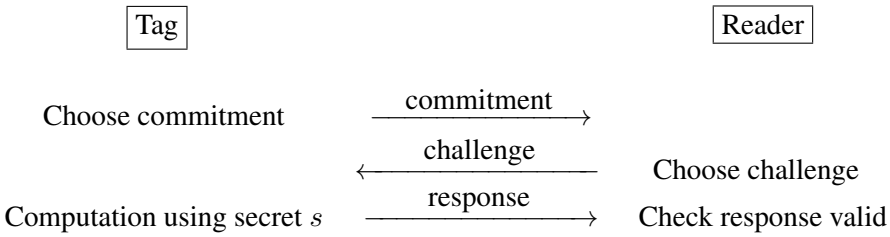
3.3 Cryptographic protocols.

We have separated algorithms from protocols since they operate at different levels. We can define a protocol for some purpose such as tag authentication, and this will consist of a communication interchange and some data transformations. The transformations could consist of a complete cryptographic algorithm, *e.g.* a block cipher or a hash function. However the technical properties of the protocol do not normally depend on *which* block cipher or *which* hash function is used, provided it is a secure one. Clearly though, when it comes to instantiation and deployment, the choice of underlying algorithm becomes vitally important.

One of the basic protocols is a simple *challenge-response* protocol which can be used, in principle, for device authentication in either a secret-key or public-key setting. Generally speaking, public or asymmetric key solutions for challenge-response can impose a considerable computational burden. However, in the symmetric key setting solutions are perfectly feasible on very limited devices. Additionally, extensions to reader and/or mutual authentication are straight-forward. The "cost" however would be a secret-key infrastructure.



The *commitment-challenge-response* protocol offers one way to provide tag authentication in a public-key setting while remaining computationally suitable even for very limited devices.



Within TAMPRES a range of security services are likely to be required and applications will dictate the most appropriate solutions.

4 Cryptographic Performance

When understanding the suitability of different cryptographic solutions, it is important to keep in mind the following five quantities.

SECURITY LEVEL. The security levels we see for typical Internet applications (from 128 through to 256 bits) are unlikely to be appropriate for passive RFID tag deployments or for nodes in sensor networks. The phrase "x-bit security level" refers to the length of the keys that would be used in a symmetric cryptosystem[1] and indicates that an adversary would have to perform 2^x operations to find the secret key used in a good symmetric cipher[2].

Since excessive security levels can harm performance it is worth taking care to get the balance right. However, it is a task of some delicacy to find the appropriate trade-off points between security and performance in computationally limited devices. For many commentators, there is a reasonable consensus that a security level of 80 bits is appropriate [75] when assets are worth protecting but the cost of devices, or their performance, is a limiting factor. To help appreciate the security this provides, the age of the universe is equivalent to 2^{80} microseconds.

[1] Equivalent strength key lengths for asymmetric systems can be readily derived [18].

[2] It is well know that time-memory trade-offs are available that give different perspectives to this headline figure of 2^x.

Some commentators have expressed concerns about *denial-of-service* attacks against RFID-based or sensor-node systems. However the cost and true benefit to an attacker in mounting such an attack is rarely analysed and such an attack is rarely realistic. Instead, the most significant threat to a cryptographic deployment in basic devices is likely to come from hardware attacks that extract secrets by exploiting what is termed *side-channel* information. While the costs and the anticipated gains for an attacker might be disputed, it is one of the goals of TAMPRES to consider this issue in some considerable detail. For more details on this issue see Section 5.

AREA. This is one of the major factors that impacts whether an algorithm can be supported on passive tags, and at what price. The area occupied by an implementation depends on (i) the specific details of the algorithm and (ii) on the security level. Having a higher security level will increase the area requirements. The standard way of measuring area is using the notion of a *gate equivalent (GE)*. For a given fabrication technology we know the physical area occupied by a boolean `nand` gate. The total physical area of the implementation can therefore be divided by the area of a `nand` gate to give the area cost in gate equivalents, GEs. The idea of representing area in this way is that an area comparison becomes somewhat independent of the fabrication technology that is used. The typical goal is that an implementation should require less than 3 000 GE, and this is attainable with the new generation of algorithms.

POWER. The second major factor for tag deployment is the power consumed by the algorithm. In passive devices this can lead to a restriction in read range while in powered devices any signficant power consumption can impact battery life. Since larger circuits consume more power[3], power and energy consumption are somewhat dependent on the chosen security level and the implementation of the algorithm. It can also be dependent on how fast we want algorithms to run, which can in turn have implications for higher-level protocols. It is notoriously difficult to compare power requirements between different fabrication technologies, but simulation tools can give a reasonable indication of the anticipated performance.

TIME. Much depends on the application whether the time to run a cryptographic computation is important. There are no guidelines on what would constitute an acceptable response time, much will depend on the application, but integrating cryptography into a communication protocol on a constrained device is a substantial step. To achieve a faster computation one could always clock the hardware faster. However this would consume more power and so it becomes a delicate issue to find the appropriate balance. This is an area of ongoing research though we observe that (i) response times for many existing techniques are already so good that the impact may be minimal and (ii) for applications that call upon cryptographic functionality, a slight reduction in tag throughput may well be an entirely appropriate trade-off.

COMMUNICATION. Some protocols in the cryptographic literature manage to have good area, power, and timing characteristics. They might even have plausible arguments for their security. But they can be let down by the amount of communication that is generated and which can become unwieldy or even infeasible.

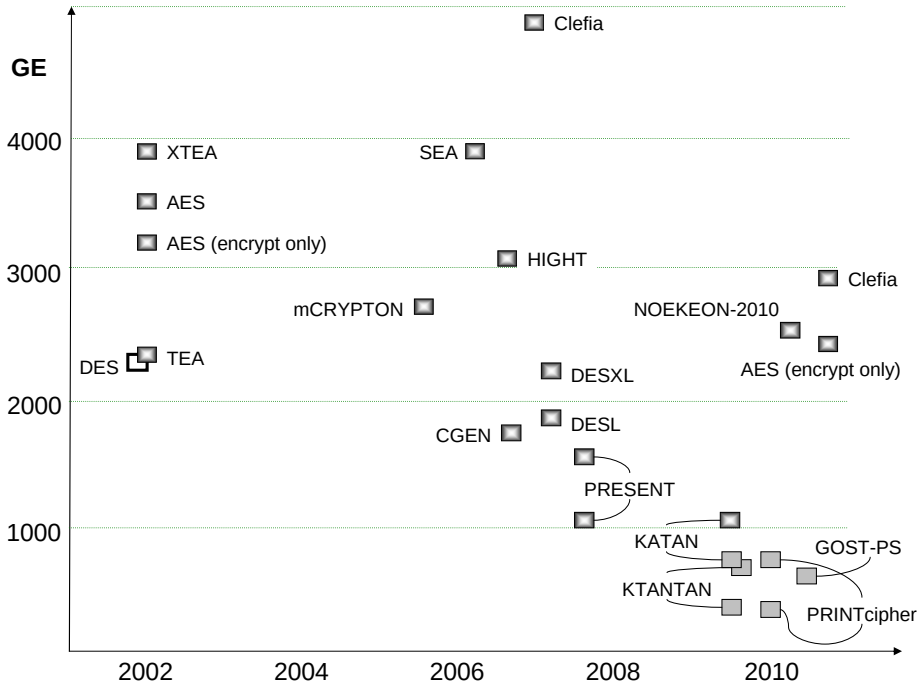
As we have noted above, these five goals can be in opposition. This means that there is a great deal of latitude to the designer/implementor in choosing their priorities. An implementation minimising execution time, for instance, will have very different area and power requirements than an implementation minimising area or power requirements. In the comparisons that follow, our focus will be on area-based comparison between algorithms. The references, however, will give more detail on the power consumption and the time impact of using the indicated algorithms.

[3]Things are more complex than this, but it is a reasonable approximation.

4.1 Symmetric algorithms

Tables 1 and 2 summarise the implementation cost of many block ciphers (except AES which is treated separately), stream ciphers, and hash functions. For message authentication codes we could use a block cipher in an appropriate mode which would carry a slight overhead over the basic block cipher. When using a block ciphers in a constrained environment, it is not clear that one needs both encryption and decryption. Provided sender and received agree on the appropriate techniques, message encryption and decryption as well as message authentication and verification can all be done using a block cipher in the "encrypt direction". There are, of course, techniques that would require sender and receiver to use different "directions", *e.g.* CBC mode, so much depends on the security architecture.

Implementations in Table 1 are, typically, optimised for area. There are, of course, many other results in the literature for, say, implementations where high throughput is required. However the area/cost of such implementations is unlikely to be suitable for constrained devices. We further note that some unconventional proposals such as HUMMINGBIRD [19] and ARMADILLO [4] have been made in the literature though attacks on early versions of both [77, 1] have lead to re-designs. The evolution of lightweight block ciphers over the past decade is illustrated below. While not permitting rigorous interpretation (data-points have not been normalised for different block and key sizes) the general trend towards the bottom right-hand corner is clear. Proposals indicated with a solid box use a fixed key, unconventionally small block size, or experimental research-oriented techniques. Few, if any, from these particular proposals remain uncompromised.



4.1.1 AES

As the NIST standard block cipher, the *Advanced Encryption Standard (AES)* [55] would be the only choice to consider for most applications. While it has a very pleasing performance profile across a wide-range of implementations, it is not known as the most compact block cipher.

There are several key results on the compact implementation of the AES, given in the following table.

ref.	*area (GE)*	*cycles/encrypt*	*bits/cycle*
[22]	3 400	1 032	0.12
[33]	3 100	160	0.80
[72]	2 400	210	0.61

Table 1: The implementation results for a variety of compact block ciphers. Throughput and efficiency are measured when the tag is clocked at a typical tag frequency of 100 KHz. The measure of efficiency used here (THROUGHPUT/AREA) means that a higher number is better.

	key size	*block size*	*cycles*	*through.* (Kbps) (*higher better*)	*efficiency* (bps/GE) (*higher better*)	*area* (GE)
fully functional						
SEA [46]	96	96	93	103.2	27.5	3,758
XTEA [38]	128	64	112	57.1	16.4	3,490
AES-128 [22]	128	128	1032	12.4	3.6	3,400
HIGHT [36]	128	64	34	188.2	61.7	3,048
Clefia [80]	128	128	176	72.7	24.3	2,996
mCrypton [45]	96	64	13	492.3	183.6	2,681
TEA [87]	128	64	64	100	42.5	2,355
DES [74]	56	64	144	44.4	19.3	2,309
KLEIN [31]	128	64	22	291	131.5	2,213
DESXL [74]	184	64	144	44.4	20.5	2,168
KLEIN [31]	80	64	16	400	190.7	2,097
KLEIN [31]	64	64	12	533.3	269.2	1,981
encryption only						
Clefia [80]	128	128	176	72.7	25.1	2,893
PRESENT-128 [11]	128	64	32	200	106.0	1,886
PRESENT-80 [11]	80	64	32	200	127.4	1,570
PRESENT-80 [76]	80	64	563	11.4	10.6	1,075
KATAN64 [14]	80	64	255	25.1	23.8	1,054
KATAN32 [14]	80	32	255	12.5	15.6	802
encryption only, fixed key						
GOST-FB [73]	256	64	32	24.2	30.3	800
PRINTcipher96 [39]	160	96	768	3.1	4.3	726
KANTAN64 [14]	80	64	255	25.1	36.5	688
GOST-PS [73]	256	64	32	24.2	37.2	651
KTANTAN32 [14]	80	32	255	12.5	27.1	462
PRINTcipher48 [39]	80	48	768	6.3	15.7	402

For very constrained devices such as UHF RFID tags, the viability of AES is not so clear at this time. However, for the applications (and devices) targetted by TAMPRES it could well be an interesting implementation choice. In addition to the implementations given above, it is therefore interesting to note that TAMPRES partners have their own implementations:

partner	*tech. (μm)*	*area (GE)*	*cycles/encrypt*	*bits/cycle*
NXP	0.14	3 162	1 060	0.12
IHP	0.25	8 700	60	2.10

4.1.2 PRESENT and Grain

We illustrate the current state-of-the-art for lightweight block and stream ciphers by comparing PRESENT with the Grain family. The set of Grain algorithms were initially developed as part of the EU-funded eSTREAM project [75] and they are often viewed as very promising stream cipher proposals for lightweight implementation. Both PRESENT and Grain v1 are of a similar age and have received similar scrutiny in the literature. While PRESENT is unchanged since its design and remains uncompromised by cryptanalysis [15, 16, 37, 43, 54, 63, 66, 85, 88], Grain has been updated several times. The first version, Grain, was broken [9] in the first phase of eSTREAM and Grain v1, that supports both 80- and 128-bit keys, was proposed for the second phase [35]. Recent analysis, including [17], has lead to changes to the version taking 128-bit keys and Grain-128a [2], which also supports an authentication mechanism, has recently been proposed.

Some sample implementation figures are provided for both the 80- and 128-bit key versions of PRESENT [32] and Grain v1 [32]. Estimates for the performance of Grain-128a have been given by the designers [2] and these are used in place of the 128-bit key version of Grain v1 which is no longer recommended. Power and throughput (*t'put*) are given for a frequency of 100 Khz, though power figures are illustrative since there are significant variations between hardware technologies. Since the Grain family requires initialisation, the time to encrypt 64 and 256 bits (in cycles) is given both with, and without, initialisation.

key (bits)	*init* (cycles)	*tech.* (μm)	*power* (μW)	*t'put* (Kbps)	*only encrypt* (cycles)		*init+encrypt* (cycles)		*area* (GE)
					64	256	64	256	
PRESENT									
80	0	0.18	5.0	200	32	128	32	128	1 570
80	0	0.18	3.3	200	32	128	32	128	1 623
128	0	0.18	-	200	32	128	32	128	1 886
Grain v1 *(80-bit key)*									
80	160	0.13	3.3	100	64	256	224	416	1 294
80	40	0.13	4.5	400	16	64	56	104	1 678
80	20	0.13	6.1	800	8	32	28	52	2 191
Grain-128a *(without message authentication)*									
128	320	-	-	50	128	512	448	832	2 146
128	160	-	-	100	64	256	324	416	2 243
128	80	-	-	200	32	128	112	208	2 438
128	40	-	-	400	16	64	56	104	2 828

In the project TAMPRES we have a variety of applications in mind. However, considering the state of the art of cryptanalysis for both Grain and PRESENT, TAMPRES partners feel that PRESENT would be a more appropriate solution at this time.

Table 2: The implementation results for a variety of hash functions. Throughput and efficiency are measured when the tag is clocked at a typical tag frequency of 100 KHz. The measure of efficiency used here (THROUGHPUT/AREA) means that a higher number is better.

	$\log_2$(security)			*through.*	*eff.*	*area*
	pre-image	*2nd pre-image*	*collision*	(Kbps)	(bps/GE)	(GE)
DM-PRESENT-80 [12]	64	64	32	15	9	1 600
DM-PRESENT-128 [12]	64	64	32	3	12	1 886
DM-PRESENT-80 [12]	64	64	32	242	110	2 213
DM-PRESENT-128 [12]	64	64	32	388	153	2 530
H-PRESENT-128 [12]	128	128	64	200	47	4 256
H-PRESENT-128 [12]	128	128	64	11	5	2 330
U-QUARK [3]	128	64	64	1	1	1 379
U-QUARK [3]	128	64	64	12	5	2 392
D-QUARK [3]	160	80	80	2	1	1 702
D-QUARK [3]	160	80	80	18	7	2 819
T-QUARK [3]	224	112	112	3	1	2 296
T-QUARK [3]	224	112	112	50	11	4 640
SHA-1 [64]	160	160	80	149	27	5 527
SHA-1 [64]	160	160	80	149	24	6 122
MD4 [23]	128	128	64	112	15	7 350
MD5 [24]	128	128	64	84	10	8 001
MAME [86]	256	256	128	267	33	8 100
SHA-1 [23]	160	160	80	40	5	8 120
SHA-256 [23]	256	256	128	45	4	10 868

4.2 Asymmetric algorithms

In attempts to use digital signatures, much work has focused on the efficient implementation of *elliptic curve cryptography (ECC)*. Currently, no implementation offering elliptic curves with a good security level has so far been published that comes close to 5 000 GE. Instead there are implementations with a lower security level requiring around 10 000 GE or more [8, 27]. Much depends on the applications in mind for TAMPRES whether or not elliptic curve based solutions are likely to be suitable.

Gaubatz *et al.* [28] have investigated the hardware efficiency of the NTRUencrypt algorithm [62]. Though their implementation for a version with reduced security requires only 2 850 GE, it takes a rather long 29 225 clock cycles. There do not appear to be any implementation figures for NTRUsign. A company called SecureRF [78] advertises a range of security-enhanced tags using something called an *algebraic eraser*. No details of their techniques are given. With regards to identification schemes, Oren *et al.* propose a scheme called WIPR [65]. An ASIC implementation of WIPR requires 5 705 GE and 66 048 clock cycles. However all these have a larger area and/or much longer processing time than CRYPTOGPS [30]. There have been numerous studies of its implementation efficiency in ASIC [49, 50] as well as the implementation of a fully-functioning FPGA prototype [29]. A full implementation of CRYPTOGPS has even been completed in silicon [71] and the *measured* area lies between 2 400 and 4 400 GE depending on the particular implementation trade-off. A version requiring 2 800 GE performs the necessary computation in less than 800 clock cycles.

The implementation results for a variety of asymmetric techniques are given below. These only offer a guide since different manufacturing technologies may give some variation. The results for CRYPTOGPS are *measured* results after full fabrication.

	time (cycles) *(lower better)*	*time* (ms) *(lower better)*	*area* (GE) *(lower better)*
CRYPTOGPS [71]	9 319	93	2 403
CRYPTOGPS [71]	724	7	2 876
WIPR [65]	66 048	660	5 705
Elliptic curve op. 2^{113} [8]	53 000	530	8 104
Elliptic curve op. $(2^{67})^2$ [8]	418 250	4 182	12 944
Elliptic curve op. 2^{113} [8]	75 250	752	14 735
Elliptic curve op. p_{192} [24]	502 000	5 020	23 600

While the results above demonstrate that asymmetric solutions for unilateral device authentication are available—most notably to prevent cloning and tag forgery in passive UHF RFID tags—it is likely that applications in TAMPRES will require more than just unilateral authentication. We are therefore likely to return to the issue of elliptic curve cryptography and supporting digital signatures.

4.2.1 TAMPRES partners and ECC

We illustrate the current state-of-the-art in elliptic curve lightweight hardware implementation, before presenting results from TAMPRES partners.

The comparison of different hardware elliptic curve implementation is not easy since it depends on a variety of factors:

- the implementation of the Arithmetic Logical Unit (`ALU`, or sometimes called `MALU` for Modular ALU) which implements the underlying field operations and, in turn, depends on:
  - the choice and representation of the underlying field, and
  - whether it supports only one or multiple fields. An `ALU` for one specific field with specified representation will be generally better in terms of space and time than a general one.
- whether it supports only one or multiple curve(s),
- the choice of the coordinate system,

Table 3: The implementation results for a variety of EC multipliers for binary curve with underlying field $\mathbb{F}_{2^{163}}$. The digit size is the size in bits of the basic word used for the field multiplication, kGE is the area of the multiplier expressed in Kilo Gate Equivalent, kCycles is the number of kilocycles needed for one scalar multiplication, the frequency indicated is the one of the block and the power is expressed for one scalar multiplication.

Implementation	Digit size	kGE	kCycles	Frequency	Power (μW)
EC Multiplier $\mathbb{F}_{2^{163}}$[10]	1	10,392	-	847 kHz	46
	4	12,876		847 kHz	79
	8	16,247		847 kHz	126
EC Multiplier $\mathbb{F}_{2^{163}}$[34]	16	11,904	296	847,5 kHz	67.23
EC Multiplier $\mathbb{F}_{2^{163}}$[42]	1	15,094	430,654	13,56 Mhz	-
[42] with extra register	1	16,206	376,864	13,56 Mhz	-
EC Multiplier $\mathbb{F}_{2^{163}}$[44] type 1	1	12,506	302,457	1,130kHz	36,6289
	4	15,356	105,183	323 kHz	12,0758
[44] type 2	1	13,624	298,111	1,130 kHz	38,7183
	4	16,433	100,837	301 kHz	12,7791
[44] type 3	1	14,307	293,587	1,130 kHz	43,4442
	4	19,693	96,311	283 kHz	13,2387
EC Multiplier $\mathbb{F}_{2^{163}}$[40] on Edward Curve	1	12,11	1499,716	400 kHz	8,12
	4	14,074	505,975	400 kHz	11,36
EC Multiplier $\mathbb{F}_{2^{163}}$[41] on Edward Curve	1	11,720	219,148	400 kHz	7,27
	4	13,427	59,800	400 kHz	11,997

- the choice of the scalar multiplier, and
- the choice of the full elliptic curve cryptographic algorithm used.

Most of the optimized implementations of elliptic curve cryptography use the combination of these different units optimized one by one and as a whole. For example, one can implement optimized scalar multiplier algorithm for a specific type of coordinate. One can also avoid some inversion operations in the ECC algorithm by only using x coordinates of the resulting elliptic curve point. To synthesize the state of the art, we choose to report only one type of implementation and this is given in Table 3. We believe this to be a good representative, and it has following characteristics:

- The implementations use fixed underlying field $\mathbb{F}_{2^{163}}$ with fixed representation. This size of field offers a good trade-off between security level and costs.
- The curve parameters and the coordinate system are fixed for one implementation, but can differ from one to another.
- We only compare the scalar multiplier. The cost in power and time is expressed for one scalar multiplication.

A full hardware lightweight implementation of elliptic curve cryptography state-of-the-art is a specific deliverable from project TAMPRES.

5 Side-Channel Resistant Implementations

The challenge in the TAMPRES project is to implement algorithms in a side-channel secure way. In particular, the crypto cores are required to be able to deal with passive attacks like power analysis attacks as well as with active attacks like fault injection attacks.

Side-channel attacks based on instantaneous power consumption exploit the fact that the power drain of the circuit depends on the internally processed values which in turn depends on the secret key material processed inside the device. These attacks can be further divided into the classes of simple and differential attacks. The first class covers attacks which use the power consumption traces of a single or a few executions of the algorithm and have been most prominently applied against asymmetric primitives. Against symmetric primitives, where the key is incorporated much faster, usually differential side-channel attacks are performed. That is, during the attacks, a part of the key (a so-called sub-key) is guessed and then for this fixed sub-key, the power consumption is predicted for a large number of inputs, typically hundreds to millions. For the same inputs, the power consumption of the real device is recorded. Afterwards, a statistical test is applied in order to compare the predicted power consumption and the real power consumption. Each comparison yields a score value for a key guess and finally the highest amongst all the score values indicates the most likely sub-key. Ways to counteract these attacks rely on either keeping the power consumption constant or random.

Fault attacks are active attacks and therefore alter the state of the device before or during an encryption. Such a corrupted state results in an erroneous output which might then be used to reduce the key entropy. In general counteracting fault attacks requires the deployment of some form of redundancy which allows detecting a corrupted state.

Both families of attacks are amongst the most practical to cryptographic devices nowadays. Therefore, countermeasures are vital for the security of an implementation. At the same time such countermeasures severely impact either the area or the execution time and hence also the energy consumption of an implementation.

Within the field of lightweight cryptography, side-channel resistance is only just beginning to be addressed in any significant way. And while both symmetric and asymmetric cryptosystems are, in principle, vulnerable to these physical attacks, the literature for the symmetric case is much better developed. The primary reason for this being the fact that only very few asymmetric techniques have the necessary performance profile to interest researchers in lightweight cryptography in the first place. We therefore motivate the field of study by referring primarily to the implementation of symmetric cryptography.

Hardware implementations usually deploy a mixture of hiding countermeasures, masking and secure logic styles. Implementations which rely only on architectural hiding countermeasures are the most inexpensive in terms of area but also show the least side-channel resistance. Common practices for hiding are shuffling or parallelization of which the latter one is more appealing in hardware. However, to achieve a sufficient level of security, also masking has to be applied. In general, it has been shown that the side-channel resistance of an implementation is proportional to SRN^d where the SNR is the signal-to-noise ratio and d is the number of shares used by the masking scheme [82]. That is, higher order masking schemes can provide a high level of security as the impact of them grows exponentially with d. On the other hand implementing sound masking schemes in hardware is a complex endeavor. Especially, effects like glitches have shown to compromise many hardware masking schemes in the past (.e.g. [48]).

To this day, only first order masking has been soundly implemented in hardware. Following the proposal of Nikova et al. [61], for instance the block cipher PRESENT has been implemented [72]. The scheme is well suited for ciphers using small S-boxes and thus the area increase lies between factors of 2.3 and 3.5, depending on the security options. An FPGA implementation of the design has been evaluated and using CPA, 1M traces were needed to recover the key. By introducing some additional noise, 5M traces were not sufficient. Recently, also AES has been implemented using this approach [53]. However, since AES uses an 8-bit S-box and implementing such a complex function using Nikova's approach is still an unsolved problem, the authors refresh the masks during the S-box computation. This requires 44 random bits per cycle. Note, that the circuit for the generation of these bits is not included in the area of 11kGE. The area increase is mostly due to the additional 256-bit mask registers and the 20 times larger S-box. The number of required traces needed to mount a successful attack increased to 75 million using MIA.

Alternatively, secure logic styles have been proposed to deal with the issue of side-channel leakage. The family of proposed logic styles is vast and we therefore focus only on one design which is based on a hiding

Table 4: Comparison of side-channel protected block cipher implementations.

Implementation	Area	Cycles	Evaluation Results
[72]	2.3-3.5kGE	547	CPA >5M traces
[53]	>11kGE	266	MIA 75M traces
[83]	$\times$ 3.5	11	CPA >1.5M traces
[69]	$\times$ 5	-	>3.5M traces
[26]	19.5kGE	2081	$\times$1700

logic style, namely WDDL [84], and one which is based on a masked logic style, namely MDPL [70]. In the first case, the design is a high performance AES which needs 11 cycles per block. The area increase due to WDDL is 3.5 times and DPA has not been successful using 1.5M traces [83]. For MDPL, the area increase is 5 times and the power consumption increase is 11 times [69]. Using, PDF attacks for mask biasing and CPA, the key could not be recovered with 3.5M traces. Although here, the countermeasure seems to be effective, it should be noted that MDPL has a so-called early propagation issue and can turn out to be very weak depending on the design. Fixing this issue led to iMDPL, the area increase of which is 20 times.

Alternatively, to save area of secure logic style and to overcome the masking issue, hybrid approaches have been pursued. Essentially, they protect the linear parts by masking and shuffling and the non-linear parts by secure logic styles. One such example is described in [26]. The area for the complete AES design increases by a factor of 6.5 which is due to the large factor inherited from iMDPL. Using a different logic style could decrease the area significantly. The increase of the number of needed traces is by a factor of 1700.

For fault attacks, countermeasures are either based on modular redundancy in various granularities or on error detection codes. For the first one, often the fact that block ciphers are bijections is used. That is, the result of decrypting the ciphertext is compared to the original input. Such a countermeasure offers a large design space. By for instance implementing it on a round basis within an iterated block cipher, the error-detection latency can be decreased and the throughput increased while the additional hardware overhead can only be moderate. Other proposals suggest to rely on error detection codes. However, in hardware, the assumption of the attacker, has a significant impact on the area increase. As the decision is often made in favor of area, it seems to be the safest choice to explore time redundant approaches. A good overview can be found in [47].

Eventually, we are also not aware of any publication, except for protocol level approaches, which deals with both problems at the same time. One such approach would be for instance [51].

Summarizing, we can draw two conclusions. First, for low-cost ciphers, Nikova's masking scheme is an appealing solution. Thus, if PRESENT is chosen for one of the cores, this would be the favorable countermeasure against side-channel attacks. Furthermore, due to the lower complexity of the PRESENT S-box (compared to the AES Rijndael's one), also coding techniques might be appealing to protect against fault analysis. Second, for block ciphers with more complex S-boxes, the costs of countermeasures are rather high, especially when thinking of combinations of fault and side-channel countermeasures. In view of this, it looks more appealing to rely on protocol level countermeasures and recently studied leakage resilient constructions [81]. Especially, because the use of protocol-level countermeasures allows an adaption of the security level and an independent overhead for side-channel and fault countermeasures.

While our focus in this section has been on symmetric cryptography, there are—as previously stated—parallel concerns for asymmetric algorithms. Within the field of elliptic curve cryptography, see Section 4.2, some of the concerns of side-channel cryptanalysis can be countered in the following ways.

When performing elliptic curve computations, the physical behavior of any executed sequence of operations should not depend on the value of a (secret) multiplier; in this way any elliptic point computation would be inherently protected against simple power analysis and timing attacks. However, hardware countermeasures are needed to hide traces of any conditional execution, *i.e.* it should not be obvious, for instance from the power profile, to see which execution path has been followed after an `if`-statement.

Similarly to symmetric cryptography, secure logic styles have been proposed to deal with the issue of side-channel leakage for elliptic curve cryptosystems. Further, again as for symmetric cryptographic, techniques against fault attacks might either repeat all or part of the cryptographic computation (or its inverse operation) or perhaps introduce invariants to the implementation that must hold during the computation. When the cryptographic operation is done, the implementation can check whether the invariant is still valid. Another technique

is to use error detection codes and to regularly check for correctness of certain values. As well as protecting elliptic curve parameters against errors in the long term non-volatile system as well as the stored secret key, a scheme for error detection and verification of data prior to scalar multiplication might conceivably be used.

The mathematical foundations of elliptic curve cryptography can permit (at least in theory) some very specific attacks, often involving points of unusually small order. Checks to ensure that the input is on the predefined curve will help, which will also prevent so-called *weak* and *twised* curve attacks. Resistance against differential power analysis can be achieved by randomizing the input point (or its representation) since such randomization destroys the correlation between the scalar, the input or output of the computation, and intermediate results of the scalar computation.

In summary, a range of countermeasures exist, each suitable for different potential attacks, but they tyically carry some implementation cost. It is therefore an open, and pressing, issue to decide which countermeasures are appropriate against which level of adversary, particularly when the target devices are highly constrained as is the case for sensor networks.

6 Cryptographic Algorithms and TAMPRES

- Symmetric techniques: AES as a byte-oriented design and PRESENT as a nibble-oriented design.
- Asymmetric techniques: ECC cryptography.

References

[1] M. Abdelraheem, C. Blondeau, M. Naya-Plasencia, M. Videau, and E. Zenner. Cryptanalysis of Armadillo-2. Available via eprint.iacr.org/2011/160.pdf.

[2] M. Ågren, M. Hell, T. Johansson, and W. Meier. A New Version of Grain-128 with Authentication. Presented at SKEW 2011, available via skew2011.mat.dtu.dk/proceedings/.

[3] J.-P. Aumasson, L. Henzen, W. Meier, and María Naya-Plasencia. Quark: A Lightweight Hash. In S. Mangard and F.-X. Standaert, editors, Proceedings of CHES '10, volume 6225 of LNCS, pages 1–15, Springer, 2010.

[4] S. Badel, N. Dagtekin, J. Nakahara, K.. Ouafi, N. Reffé, P. Sepehrdad, P. Susil, and S. Vaudenay. Armadillo: A multi-purpose cryptographic primitive dedicated to hardware. In S. Mangard and F.-X. Standaert, editors, Proceedings of CHES '10, volume 6225 of LNCS, pages 398–412, Springer, 2010.

[5] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita. Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis. In D. Stinson and S. Tavares, editors, *Proceedings of SAC '00*, volume 2012 of LNCS, pages 39–56. Springer-Verlag, 2000.

[6] D. Arditti, C. Berbain, O. Billet, and H. Gilbert. Compact FPGA Implementations of QUAD. In F. Bao and S. Miller, editors, *Proceedings of ASIACCS 2007*. ACM Press, 2007.

[7] S. Babbage and M. Dodd. The MICKEY Stream Ciphers. In M. Robshaw and O. Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 191–209. Springer, 2008.

[8] L. Batina, J. Guajardo, B. Preneel, P. Tuyls, and I. Verbauwhede. Public-Key Cryptography for RFID Tags and Applications. In P. Kitsos and Y. Zhang, editors, *RFID Security: Techniques, Protocols and System-on-Chip Design*, pages 317–348. Springer, 2008.

[9] C. Berbain, H. Gilbert, and A. Maximov. Cryptanalysis of Grain. In M. Robshaw, editor, *Proceedings of FSE '06*, volume 4047 of LNCS, pages 15–29. Springer-Verlag, 2006.

[10] H. Bock,M. Braun, M. Dichtl, E. Hess, J. Heyszl, W. Kargl, H. Koroschetz, B. Meyer, H. Seuschek: A Milestone Towards RFID Products Offering Asymmetric Authentication Based on Elliptic Curve Cryptography, 2008.

[11] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In P. Paillier and I. Verbauwhede, editors, *Proceedings of CHES '07*, volume 4727 of LNCS, pages 450–466. Springer, 2007.

[12] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, and Y. Seurin. Hash Functions and RFID Tags: Mind the Gap. In E. Oswald and P. Rohatgi, editors, *Proceedings of CHES '08*, volume 5154 of LNCS, pages 283–299, Springer, 2008.

[13] C. de Cannière and B. Preneel. Trivium. In M. Robshaw and O. Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 244–266. Springer, 2008.

[14] C. de Cannière, O. Dunkelman, and M. Knezević. KATAN and KTANTAN–A Family of Small and Efficient Hardware-Oriented Block Ciphers. In C. Clavier and K. Gaj, editors, *Proceedings of CHES 2009*, volume 5747 of LNCS, pages 272–288. Springer, 2009.

[15] J.L. Cho. Linear cryptanalysis of reduced-round PRESENT. In J. Pieprzyk, editor, *Proceedings of CT-RSA 2010*, volume 5985 of LNCS, pages 302–317, Springer, 2010.

[16] B. Collard and F.-X. Standaert. A statistical saturation attack against the block cipher PRESENT. In M. Fischlin, editor, *Proceedings of CT-RSA '09*, volume 5473 of LNCS, pages 195–210, Springer, 2009.

[17] I. Dinur and A. Shamir. Breaking Grain-128 with Dynamic Cube Attacks. n A. Joux, editor, *Proceedings of FSE '11*, LNCS, Springer, to appear.

[18] ECRYPT. ECRYPT Final Report on Algorithms and Key Lengths (2008). Available via www.ecrypt.eu.org/ecrypt1/documents.html.

[19] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith. Ultra-Lightweight Cryptography for Low-Cost RFID Tags: Hummingbird Algorithm and Protocol. Available via www.cacr.math.uwaterloo.ca/techreports/2009/cacr2009-29.pdf.

[20] D. Engels, M.-J. O. Saarinen, and E. M. Smith. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. Available via eprint.iacr.org/2011/126.pdf.

[21] M. Feldhofer. Comparison of Low-Power Implementations of Trivium and Grain. *State of the Art of Stream Ciphers 2007 (SASC 2007)*, workshop record. February 2007. Available via www.ecrypt.eu.org/stream/.

[22] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems Using the AES Algorithm. In M. Joye and J.-J. Quisquater, editors, *Proceedings of CHES '04*, volume 3156 of LNCS, pages 357–370, Springer-Verlag, 2004.

[23] M. Feldhofer and C. Rechberger. A Case Against Currently Used Hash Functions in RFID Protocols. In *Proceedings of IS '06*, volume 4277 of LNCS, pages 372-381, Springer-Verlag, 2006.

[24] M. Feldhofer and J. Wolkerstorfer. Hardware Implementation of Symmetric Algorithms for RFID Security. In P. Kitsos and Y. Zhang, editors, *RFID Security: Techniques, Protocols and System-on-Chip Design*, pages 373–415. Springer, 2008.

[25] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. AES Implementation on a Grain of Sand. *Information Security, IEE Proceedings*, 152(1):13–20, 2005.

[26] M. Feldhofer and T. Popp. Power Analysis Resistant AES Implementation for Passive RFID Tags. In C. Lackner, T. Ostermann, M. Sams, and R. Spilka, editors, *Austrochip 2008*, pages 1 – 6, 2008.

[27] F. Fürbass and J. Wolkerstorfer. ECC Processor with Low Die Size for RFID Applications. In *Proceedings of The IEEE International Symposium on Circuits and Systems 2007 – ISCAS 2007*, pages 1835–1838, 2007.

[28] G. Gaubatz, J.-P. Kaps, and B. Sunar. Public key Cryptography in Sensor Networks—Revisited. In C. Castellucia, H. Hartenstein, C. Paar, and D. Westhoff, editors, *Proceeding of ESAS '04*, volume 3312 of LNCS, pages 2–18, Springer-Verlag, 2004.

[29] M. Girault, L. Juniot, and M. Robshaw. The Feasibility of On-the-Tag Public Key Cryptography. *RFIDsec 2007*, workshop record. Available via `rfidsec07.etsit.uma.es/slides/papers/paper-32.pdf`.

[30] M. Girault, G. Poupard, and J. Stern. On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order. Journal of Cryptology, vol. 19, pages 463–487, Springer, 2006.

[31] Z. Gong, S. Nikova, and Y.-W. Law. KLEIN, a new family of lightweight block ciphers. Available via `http://doc.utwente.nl/73129/`.

[32] T. Good and M. Benaissa. ASIC hardware performance. In M. Robshaw and O. Billet, editors, *New Stream Cipher Designs*, volume 4986 of LNCS, pages 267–293, Springer, 2008.

[33] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen. Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core. In *DSD*, pages 577–583, 2006.

[34] D. Hein, J. Wolkerstorfer, N. Felber : ECC is Ready for RFID A Proof in Silicon, 2008.

[35] M. Hell, T. Johansson, and W. Meier. Grain - A Stream Cipher for Constrained Environments. In M. Robshaw and O. Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 179–190, Springer, 2008.

[36] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In L. Goubin and M. Matsui, editors, *Proceedings of CHES '06*, volume 4249 of LNCS, pages 46–59, Springer-Verlag, 2006.

[37] S. Kerckhof, B. Collard, F.-X. Standaert. FPGA Implementation of a Statistical Saturation Attack against PRESENT. In *Proceedings of Africacrypt 2011*. To appear, Springer.

[38] J.-P. Kaps. Chai-tea, Cryptographic Hardware Implementations of xTEA. In D. Chowdhury, V. Rijmen, and A. Das, editors, *Proceedings of Indocrypt '08*, volume 5365 of LNCS, pages 363-375, Springer, 2008.

[39] L.R. Knudsen, G. Leander, A. Poschmann, and M.J.B. Robshaw. PRINTCIPHER: A Block Cipher for IC-Printing. In S. Mangard and F.-X. Standaert, editors, *Proceedings of CHES '10*, volume 6225 of LNCS, pages 16–32. Springer, 2010.

[40] Ü. Kobabaş, Hardware Implementations of ECC Over a Binary Edwards Curve, 2009, Master Thesis at U.K. Leuven - ESAT.

[41] Ü. Kobabaş, J. Fan, I. Verbauwhede, Implementation of Binary Edwards Curves for very-constrained devices, 2010

[42] S. Kumar, C. Paar: Are standards compliant Elliptic Curve Cryptosystems Feasible on RFID?, 2006.

[43] G. Leander. On Linear Hulls, Statistical saturation attacks, PRESENT and a cryptanalysis of PUFFIN. In K. Paterson, editor, *Proceedings of Eurocrypt 2011*, to appear, Springer, 2011.

[44] Y. Lee, K. Sakiyama, L. Batina, I. Vaerbauwhede : Elliptic Curve Based Security Processor for RFID, 2008.

[45] C. Lim and T. Korkishko. mCrypton - A Lightweight Block Cipher for Security of Low-cost RFID Tags and Sensors. In J. Song, T. Kwon, and M. Yung, editors, *Proceedings of WISA'05*, volume 3786 of LNCS, pages 243–258, Springer-Verlag, 2005.

[46] F. Mace, F.-X. Standaert, and J.-J. Quisquater. ASIC Implementations of the Block Cipher SEA for Constrained Applications. In *RFID Security — RFIDsec 2007, Workshop Record*, pages 103–114, Malaga, Spain, 2007.

[47] T. Malkin, F.-X. Standaert, and M. Yung. A comparative cost/security analysis of fault attack countermeasures. In L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, editors, *FDTC*, volume 4236 of *Lecture Notes in Computer Science*, pages 159–172. Springer, 2006.

[48] S. Mangard and K. Schramm. Pinpointing the side-channel leakage of masked aes hardware implementations. In L. Goubin and M. Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 76–90. Springer, 2006.

[49] M. McLoone and M.J.B. Robshaw. Public Key Cryptography and RFID. In M. Abe, editor, *Proceedings of CT-RSA '07*, volume 4377 of LNCS, pages 372–384, Springer, 2007.

[50] M. McLoone and M.J.B. Robshaw. New Architectures for Low-Cost Public Key Cryptography on RFID Tags. In *Proceedings of SecureComm '05*, pages 1827–1830. IEEE Computer Society Press, 2007.

[51] M. Medwed, F.-X. Standaert, J. Großschädl, and F. Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In D. J. Bernstein and T. Lange, editors, *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.

[52] A. Menezes, P. van Oorschot, and S. Vanstone. Handbook of Applied Cryptography, CRC Press, 1996.

[53] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang. Pushing the limits: A very compact and a threshold implementation of aes. In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 69–88. Springer, 2011.

[54] J. Nakahara, P. Sepehrdad, B. Zhang, and M. Wang. Linear (hull) and algebraic cryptanalysis of the block cipher PRESENT. In A. Otsuka, editor, *Proceedings of CANS '09*, volume 5888 of LNCS, pages 58–75, Springer, 2009.

[55] National Institute of Standards and Technology. FIPS 197: Advanced Encryption Standard, November 2001. Available via `csrc.nist.gov`.

[56] National Institute of Standards and Technology. Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005. Available via `csrc.nist.gov`.

[57] National Institute of Standards and Technology. FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC), July 2008. Available via `csrc.nist.gov`.

[58] National Institute of Standards and Technology. Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation. December, 2001. Available via `csrc.nist.gov`.

[59] National Institute of Standards and Technology. Cryptographic Hash Project. Information available via `csrc.nist.gov`.

[60] National Institute of Standards and Technology. FIPS 186-3: Digital Signature Standard. June, 2009. Available via `csrc.nist.gov`.

[61] S. Nikova, V. Rijmen, and M. Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptology*, 24(2):292–321, 2011.

[62] NTRU Corporation. NTRUencrypt. Available via `www.ntru.com`.

[63] K. Ohkuma. Weak keys of reduced-round PRESENT for linear cryptanalysis. In M. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Proceedings of SAC '09*, volume 5867 of LNCS, pages 249–265, Springer, 2009.

[64] M. O'Neill (née McLoone). Low-Cost SHA-1 Hash Function Architecture for RFID Tags. *RFIDSec '08*, workshop record, pages 41–51, 2008.

[65] Y. Oren and M. Feldhofer. A Low-Resource Public-Key Identification Scheme for RFID Tags and Sensor Nodes. In *Proceedings of WiSec '09*. ACM Press, 2009.

[66] O. Özen, K. Varici, C. Tezcan, and C. Kocair. Lightweight block ciphers revisited: Cryptanalysis of reduced-round PRESENT and HIGHT. In C. Boyd and J.M.G. Nieto, editors, *Proceedings of ACISP '09*, volume 5594 of LNCS, pages 90–107, Springer, 2009.

[67] K. Ouafi, R. Overbeck, and S. Vaudenay. On the Security of HB# against a Man-in-the-Middle Attack. In J. Pieprzyk, editor, *Proceedings of Asiacrypt '08*, volume 5350 of LNCS, pages 108–124. Springer, 2008.

[68] K. Ouafi and S. Vaudenay. Smashing SQUASH-0. In A. Joux, editor, *Proceedings of Eurocrypt '09*, volume 5479 of LNCS, pages 300–312. Springer, 2009.

[69] T. Popp, M. Kirschbaum, and S. Mangard. Practical attacks on masked hardware. In M. Fischlin, editor, *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 2009.

[70] T. Popp and S. Mangard. Masked dual-rail pre-charge logic: Dpa-resistance without routing constraints. In J. R. Rao and B. Sunar, editors, *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2005.

[71] A. Poschmann, M.J.B. Robshaw, F. Vater, and C. Paar. Lightweight Cryptography and RFID: Tackling the Hidden Overheads. In D. Lee and S. Hong, editors, *Proceedings of ICISC '09*, volume 5984 of LNCS, pages 129-145, Springer, 2010.

[72] A. Poschmann, A. Moradi, K. Khoo, C.-W. Lim, H. Wang, and S. Ling. Side-channel resistant crypto for less than 2,300 ge. *J. Cryptology*, 24(2):322–345, 2011.

[73] A. Poschmann, H. Wang, and S. Lin. 256-bit Standardised Crypto for 650 GE—GOST Revisited. In S. Mangard and F.-X. Standaert, editors, *Proceedings of CHES '10*, volume 6225 of LNCS, pages 219–233. Springer, 2010.

[74] A. Poschmann, G. Leander, K. Schramm, and C. Paar. New Lightweight DES Variants Suited for RFID Applications. In A. Biryukov, editor, *Proceedings of FSE '07*, volume 4593 of LNCS, pages 196–210. Springer-Verlag, 2007.

[75] M.J.B. Robshaw. The eSTREAM Project. In M. Robshaw and O. Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 1–6, Springer, 2008.

[76] C. Rolfes, A. Poschmann, G. Leander, and C. Paar. Ultra-Lightweight Implementations for Smart Devices – Security for 1000 Gate Equivalents. In G. Grimaud, F.-X. Standaert, editors, *Proceedings of CARDIS '08*, volume 5189 of LNCS, pages 89–103, Springer, 2008.

[77] M.-J. O. Saarinen. Cryptanalysis of Hummingbird-1. In A. Joux, editor, *Proceedings of FSE 2011*, volume 6733 of LNCS, pages 328–341, Springer, 2011.

[78] SecureRF. Company information available at `www.securerf.com`.

[79] A. Shamir. SQUASH – A New MAC With Provable Security Properties for Highly Constrained Devices Such As RFID Tags. In K. Nyberg, editor, *Proceedings of FSE '08*, volume 5086 of LNCS, pages 144–157. Springer, 2008.

[80] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata. The 128-bit Blockcipher CLEFIA. In A. Biryukov, editor, *Proceedings of FSE 2007*, volume 4593 of LNCS, pages 181–195, Springer, 2007.

[81] F.-X. Standaert, O. Pereira, Y. Yu, J.-J. Quisquater, M. Yung, and E. Oswald. Leakage resilient cryptography in practice. Cryptology ePrint Archive, Report 2009/341, 2009. http://eprint.iacr.org/.

[82] F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, and S. Mangard. The world is not enough: Another look on second-order DPA. In M. Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.

[83] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede. A side-channel leakage free coprocessor IC in 0.18μm CMOS for embedded AES-based cryptographic and biometric processing. In W. H. J. Jr., G. Martin, and A. B. Kahng, editors, *DAC*, pages 222–227. ACM, 2005.

[84] K. Tiri and I. Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *DATE*, pages 246–251. IEEE Computer Society, 2004.

[85] M. Wang. Differential cryptanalysis of reduced-round PRESENT. In S. Vaudenay, editor, *Proceedings of Africacrypt '08*, volume 5023 of LNCS, pages 40–49, Springer, 2008.

[86] H. Yoshida, D. Watanabe, K. Okeya, J. Kitahara, J. Wu, O. Kucuk, and B. Preneel. MAME: A Compression Function With Reduced Hardware Requirements. In P. Paillier and I. Verbauwhede, editors, *Proceedings of CHES '07*, volume 4727 of LNCS, pages 148–165. Springer, 2007.

[87] Y. Yu, Y. Yang, Y. Fan, and H. Min. Security Scheme for RFID Tag. Auto-ID Labs white paper WP-HARDWARE-022. Available via `www.autoidlabs.org`.

[88] M. Z'aba, H. Raddum, M. Henricksen, and E. Dawson. Bit-pattern based integral attack. In K. Nyberg, editor, *Proceedings of FSE '08*, volume 5086 of LNCS, pages 363–381, Springer, 2008.

