Cryptographic Schemes Based on the ASASA Structure: Black-box, White-box, and Public-key

Alex Biryukov, Charles Bouillaguet, and Dmitry Khovratovich

University of Luxembourg and University of Lille
{alex.biryukov,dmitry.khovratovich}@uni.lu,charles.bouillaguet@univ-lille1.fr

Abstract. In this paper we pick up an old challenge to design public key or white-box constructions from symmetric cipher components. We design several encryption schemes based on the ASASA structure ranging from fast and generic symmetric ciphers to compact public key and white-box constructions based on generic affine transformations combined with specially designed low degree non-linear layers. While explaining our design process we show several instructive attacks on the weaker variants of our schemes.
Keywords: ASASA, multivariate cryptography, white-box cryptography, cryptanalysis, algebraic, symmetric.

1 Introduction

Since the development of public key cryptography in the late 1970's it has been an open challenge to diversify the set of problems on which such primitives were built as well as to find faster alternatives, since most public key schemes were several orders of magnitude slower than symmetric ones. One of the directions was to design public key schemes from symmetric components. As public key encryption requires trapdoors, they have been hidden in secret affine layers [59], field representations [65], biased S-boxes and round functions [69, 72, 80], structure of connection polynomials in stream ciphers [2]; however most of these schemes were broken [45, 52, 64, 77]. We recall that a typical symmetric cipher is built from layers of affine transformations (A) and S-boxes (S), a design principle dating back to Shannon [75]. It is thus natural to see what designs can be made from such components. Whereas the classical cipher AES-128 consists of 10 rounds with 19 layers in total [63], it is striking that a lot of effort has been put into designing public-key schemes with only 3 layers, using the ASA (affine-substitution-affine) structure. This has indeed been the mainstream of what is known as *multivariate cryptography.* However, in this case, the non-linear layer is usually an ad-hoc monolithic function over the full state, as opposed to an array of independent S-boxes.

It has been known that the scheme SASAS with two affine and three nonlinear layers is vulnerable to a structural attack if the nonlinear layer consists of several independent S-boxes [13]. The scheme ASA, though secure for a random monolithic S-box, has been shown weak in concrete multivariate proposals. In the seemingly unrelated area of *white-box cryptography* the ASA approach to build obfuscated lookup tables failed multiple times. This suggests exploring the shortest scheme unbroken so far — the ASASA construction with injective S-boxes — in the application to symmetric (black-box), public-key, and white-box cryptography. Let us overview the related areas.

Retrospective of multivariate cryptography

The idea of multivariate cryptography dates back to the Shannon's idea [75] that recovering the secrets in any cryptographic scheme could be reduced to solving particular systems of (boolean) equations. Since nearly all forms of cryptology *implicitly* rely on the hardness of solving some kind of equation systems, then it must be possible to design cryptographic schemes that *explicitly* rely on the hardness of this problem. In multivariate public-key schemes, the public-key itself is a system of polynomial equations in several variables. It is well-known that solving such systems is NP-hard, even when the polynomials are quadratic (hence the name of the MQ problem, which stands for Multivariate Quadratic polynomial systems). An additional advantage of the MQ cryptosystems is that they seem invulnerable to quantum algorithms and hence are candidates for Post-Quantum cryptography.

Multivariate polynomials have been used in cryptography in the early 1980's with the purpose of designing RSA variants with faster decryption. At this time, Imai and Matsumoto designed the first public-key scheme explicitly based on the hardness of MQ. It made it to the general crypto community a few years later under the name C^* [59].

Several years later, in 1995, Patarin [64] found a devastating attack against C^*, allowing to decrypt and to forge signatures very efficiently. Thereafter many multivariate scheme have been proposed (we counted at least 20 of them), including a plethora of bogus and vainly complicated proposal with a short lifespan. A few constructions stood out and received more attention than the others because of their simplicity and their elegance, such as HFE [65], SFLASH [66] and UOV [51].

However, the practical break of the first HFE challenge, supposed to offer 80 bits of security in 2003 [38], and the demise of SFLASH in 2007 [33], just after the NESSIE consortium proposed it to be standardized, shattered the hopes and trust of the cryptographic community at large in multivariate cryptography. This brought the multivariate fashion to a stop.

The main problem in multivariate crypto is that the selection of candidates for the nonlinear layer S is scarce (we will discuss this in Section 4). What remains usually has so strong a structure within, that it can be detected and exploited even in the presence of unknown A layers. In the last years, a few researchers started designing public-key schemes based on the hardness of *random instances* of the MQ problem [47, 73], though no drop-in replacement for conventional public-key encryption schemes has been proposed. Still, they are promising because there is a concensus that random instances are hard, and all known algorithms are exponential and impractical on random systems.

This overview clearly indicates the need of a larger structure for multivariate cryptosystems, and suggests truly random polynomials in this context, which we use in our schemes.

Retrospective of white-box cryptography

In a parallel development a notion of *white-box cryptography* (WBC) has been introduced in [24]. The initial motivation was to embed symmetric secret keys into the implementation of popular standards like AES or DES in a way that binds the attacker to the specific implementation for DRM purposes. Several proposals have been made [22, 23] with the main idea to obfuscate key-dependent parts of the cipher and publish them as lookup tables, so that the entire encryption routine becomes just a sequence of table lookups. The obfuscation constitutes of wrapping the nonlinear transformation (S) with random affine transformations (A) so that the affine layers would cancel each other after composition.

As a result, the lookup tables are just instantiations of the ASA structure. Moreover, since the nonlinear layers of AES and DES consist of independent S-boxes, the resulting ASA structure is very weak and can be attacked by a number of methods [11, 43, 61, 62]. As demonstrated by Biryukov and Shamir [13], even as large structure as SASAS is weak if the S-layers consist of smaller S-boxes. Surprisingly overlooked by the designers of white-box schemes, the generic attack [13] exploits multiset and differential properties of SASAS and applies to all the published white-box proposals so far. It appears that the mainstream ciphers are just poor choice for white-box implementations due to high diffusion properties and the way how the key is injected.

To formalize the problem, two notions have been suggested [74, 78]. The **weak white-box implementation** of a cryptographic primitive protects the key and its derivatives i.e. aims to prevent the *key-recovery attack*. This ensures that unauthorized users can not obtain any compact information (e.g. the key or the set of subkeys) to decrypt the protected content.

The **strong white-box implementation** of a primitive protects from the *plaintext-recovery attack*, i.e. does not allow to decrypt given the encryption routine with the embedded key. Such an implementation may replace the public-key cryptosystems in many applications, in particular if it is based on an existing symmetric cipher and is reasonably fast for a legitimate user. The existing white-box implementations of AES and DES [23, 24] do not comply with in this notion, since they are easily invertible, which is strikingly different from the black-box implementations of these ciphers. So far the only proposed candidate is the pairing-based obfuscator scheme with poor performance [74].

The ASASA-based designs may not only hide the key for the weak white-box implementation, but also provide non-invertibility aiming for the strong white-box construction.

Our contributions

We continue to explore the design space of compact schemes built from layers of affine mappings and S-boxes. We first note that there is no known generic attack on the 5-layered ASASA scheme with injective S-boxes in the flavour of [13], which makes the ASASA structure a promising framework for future white-box, black-box, and public-key schemes. Based on this principle, we propose and analyze the following constructions in this paper:

- Two *public-key / strong white-box* variants of the ASASA symmetric scheme: one is based on Daemen's quadratic S-boxes [26] (previously used in various hash functions) and another based on random expanding S-boxes. (Section 2). We explore standard cryptanalytic attacks such as differential, linear and others, the recent decomposition attacks [40, 41], and a new interpolation attack on weakened variants of our schemes (Section 3). We demonstrate that our set of parameters offers a comfortable security margin.
- A concrete instantiation for a fast symmetric ASASA-based blockcipher with secret S-boxes and affine layers and comparable with AES in it's encryption/decryption speed (Section 4).
- A concept of *memory-hard white-box implementation* for a symmetric blockcipher and a concrete family of ciphers with tunable memory requirements (Section 5). It prevents key recovery and requires the adversary to share the entire set of lookup tables to allow an unauthorized user to decrypt. Therefore, the cipher solves the problem of *weak white-box* implementation.

Related work. The idea of the ASASA structure dates back to [67]. The scheme named "2R" (which stands for "2 Rounds") had non-injective S-boxes and was subsequently broken in [9, 39, 79]. Modified versions of 2R have been later attacked by decomposition algorithms [40, 41], and no countermeasure has been suggested. The idea of adding perturbation, or noise, to the output of ASA-type MQ-scheme has been proposed by Patarin et al. in the design of C^*_+ [68], and later by Ding [30]. It is also related to the LWE (Learning with Error) problem, extensively discussed in [71] and used in an MQ-based hybrid encryption scheme [47].

In Section 5 we design a weak white-box-ready blockcipher out of small components, i.e. block ciphers with small block size. Ciphers with small blocks were also explored within shuffling-based schemes by Granboulan and Pornin [44], and later by Hoang et al. [46]. Ciphers working on arbitrary (possibly small) domains are often offered for format-preserving encryption [5].

2 Asymmetric ASASA schemes: strong white-box and public-key

The first ASASA cryptosystem, designed by Patarin and Goubin, was a public-key scheme with non-bijective S-boxes and was easily broken by Biham, exploiting this property in [9]. Shortly afterwards, Biryukov and Shamir explored multi-layer schemes with bijective S-boxes and demonstrated a generic attack on the structure SASAS with two affine layers [13]. The outer S-boxes are recovered with a variant of the Square attack, whereas the inner affine layers are peeled off with linear algebra methods. It was clearly demonstrated that these properties disappear in larger schemes, and no attack on ASASA or other larger structures has been proposed since.

2.1 Strong white-box security

We start with the notion of the strong white-box security that summarizes the discussion in [78].

Definition 1. *Let the pair of algorithms (E, D) be a private-key encryption scheme, which takes key K as parameter. We say that $\mathcal{O}_{E_K}$ is a* secure strong white-box implementation *for E_K if it is computationally hard to obtain $\mathcal{D}'$ equivalent to D_K given full access to $\mathcal{O}_{E_K}$.*

In other words, an adversary should be unable to decrypt given the white-box implementation $\mathcal{O}_{E_K}$ of E_K. This notion closely resembles the definition of a trapdoor permutation [50] used to construct a public-key encryption scheme. As we see, our asymmetric proposals are suitable for both notions.

2.2 Outline

We propose several asymmetric instantiations of the ASASA structure, which may serve both in the white-box and public-key setting. We have not found any reasonable use for lookup tables in this framework[1] and hence look for polynomial-based S-boxes. In order to keep the reasonable size of the description, we restrict to polynomials of degree two over some finite field, so that the resulting scheme has degree four. This approach bring us to the area of *multivariate cryptography*, which aims to design cryptographic primitives based on multivariate polynomials over finite field.

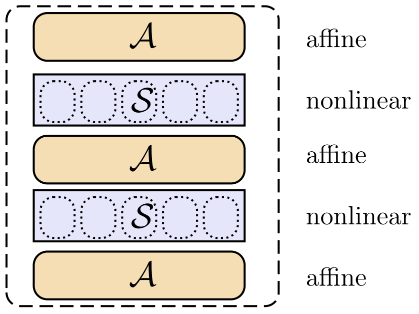

Fig. 1. The ASASA structure: two nonlinear layers surrounded by affine layers

Let us introduce the following notations. The public key/white-box implementation is exposed as a set of polynomials $\mathbf{b}$, which is constructed out of the following composition:

$$\boxed{\mathbf{b} = \mathcal{U} \circ \mathbf{a}_2 \circ \mathcal{T} \circ \mathbf{a}_1 \circ \mathcal{S},} \tag{1}$$

where $\mathbf{a}_1, \mathbf{a}_2$ are nonlinear transformations, and $\mathcal{U}, \mathcal{T}, \mathcal{S}$ are affine transformations.

There have been many proposals for nonlinear layers in the ASA structure, and various attacks exploited these choices. Most attacks are not evidently translated into degree 4, as they compute, e.g., differentials of the public key, which appear to be linear in the ASA case. The notable exception is the decomposition attack [40, 41], that will be discussed in Section 2.3.

We offer two fresh ideas for the nonlinear layers in ASASA. The first candidate is the so called χ-function. It derives from invertible cellular automata and was brought into symmetric cryptography by Daemen. To the best of our knowledge, it has never been used in multivariate cryptography.

The second candidate is a set of random injective S-boxes of degree 2. Since the families of low degree permutations are small and do not absorb much randomness, we propose to use expanding S-boxes, which can be key-dependent and generated online. Having the expansion rate of 2, it is rather easy to obtain injective transformations and still keep them quadratic[2].

Limitations for expanding schemes. Whatever construction is used, an expanding scheme has a clear limitation in the public-key and white-box setting. It implies that only a tiny subset of potential ciphertexts is decryptable, which makes the encryption and decryption process non-interchangeable. As a result, the expanding scheme can be used for encryption only and can not produce signatures. Also in the white-box context, it can not be used for decrypting the content. On the other hand, it can still be used to ensure tamper-resistance of software [60].

2.3 Defeating decomposition algorithm with perturbations

The authors of recently published decomposition algorithms [40, 41] claim to break ASASA schemes with quadratic nonlinear layers with complexity $O(n^9)$, where n is the number of variables. The

[1] So far all attempts to hide a trapdoor in lookup table-based designs failed. We investigated this problem and conjecture that such scheme just does not exist, at least given the state-of-the-art in the design of preimage-resistant functions.
[2] Our experiments show that S-boxes with an even smaller rate of 1.75 can be found.

decomposition problem is formulated as follows: given a set of polynomials $h = (h_1, \dots, h_u)$ over polynomial ring $\mathbb{K}[x_1, \dots, x_n]$ ($\mathbb{K}$ denoting an arbitrary field) find any $f = (f_1, \dots, f_u)$ and $g = (g_1, \dots, g_n)$ over $\mathbb{K}[x_1, \dots, x_n]$ whose composition is equal to h:

$$h = (h_1, \dots, h_u) = (f_1(g_1, \dots, g_n), \dots, f_u(g_1, \dots, g_n)).$$

In the context of the ASASA structure with quadratic S-boxes, the sets f and g, that are produced by a decomposition algorithm, are linearly equivalent to the internal ASA structures. This does not fully constitute a break, since the adversary still needs to invert both ASA constructions. The proposed algorithms also have not been applied to the parameters and fields that we choose. Nevertheless, it is desirable to find some countermeasure.

Our idea is to introduce some perturbation just after the second S layer in the form of several key-dependent secret polynomials of degree 4 (Figure 2). A similar approach has been used by Ding in his modification of C* [30] and HFE [31], and by Bringer et al. [19] in their modification of the traceable block cipher. In some cases (notably HFE), the "perturbation" would be identified and removed [34], thanks to a differential attack exploiting properties of the non-linear transformations. The use of perturbation polynomials has been also linked to the LWE (Learning with Error) framework in [47], but the full application of LWE to multivariate cryptography is still to be explored in the future.

Denoting the perturbation polynomials as another nonlinear transformation $\mathbf{a}_p$ we obtain the modified public key $\mathbf{b}_p$:

$$\mathbf{b}_p = \mathcal{U} \circ [\mathbf{a}_p + (\mathbf{a}_2 \circ \mathcal{T} \circ \mathbf{a}_1 \circ \mathcal{S})], \tag{2}$$

so

$$\mathbf{b}_p(x) = \mathbf{b}(x) + \mathcal{U}\mathbf{a}_p(x).$$

Hence the perturbation polynomials are mixed by the last affine transformation and spread over the public key. The encryption process remains exactly the same, while for decryption we have to guess the values of these polynomials. Suppose that we work over $\mathbb{F}_2$ so that $\mathbf{a}_p$ is sparse and contains only w polynomials with average Hamming weight $q \cdot 2^n$ (i.e. they are equal to 0 with probability $(1-q)$). Then the noise on average consists of qw bit flips, and we guess their positions after about $\sum_{i=0}^{qw} \binom{w}{i}$ attempts. For instance, $\mathbf{a}_p$ with 8 non-zero polynomials of weight $\approx 2^{n-1}$ requires 2^7 trial decryptions on average.

We distinguish true plaintexts from false ones either by recomputing the perturbation polynomials or by using expanding S-boxes so that noisy bits prohibit inversion. Padding the plaintexts with zero bits also helps but disallows turning encryption to decryption. The position of noisy bits does not matter much, since it would be concealed by the affine transformation. However, if we filter out noise with expanding S-boxes, it makes sense to spread the noisy bits so that an S-box can still be inverted in the presence of noise.

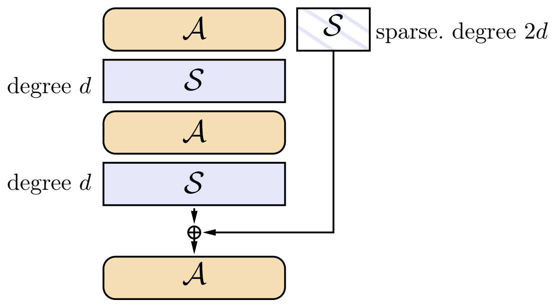

Fig. 2. Small perturbations to defeat decomposition attacks as injection of sparse high-degree polynomials

2.4 χ-scheme

Our first idea was to build the nonlinear transformation out of a popular quadratic S-box χ [26, Section 6.6.2], which has been used in hash the functions Panama [27], RadioGatun [7], Keccak [8]. The transformation χ can be defined for every odd length $k = 2t + 1$ and has the following features:

- It has degree 2 in the forward direction, but degree $(t + 1)$ in the backward direction.
- It can be efficiently inverted for every size (Appendix D).
- Its differential and linear properties have been widely studied [26].

The S-box χ of length k is defined as follows:

$$\chi(x_0, x_1, x_2, \ldots, x_{k-1}) = (y_0, y_1, y_2, \ldots, y_{k-1}),$$

where

$$y_i = x_i \oplus x_{i+1}x_{i+2} \oplus x_{i+2},$$

and indices are computed modulo k.

Regardless of the χ length (and hence the size of the S-box), we can formulate properties of the whole scheme and its features:

1. For the standard block size of 128 bits, we get approximately (since the S-box size might not divide 128) 2^7 input variables. Thus each output coordinate of $\mathbf{b}$ is a polynomial of degree 4 of 2^7 variables, has about $\binom{2^7}{4} \approx 2^{24.5}$ terms, so the full scheme description is about $2^{24.5+7-3} = 2^{28.5}$ bytes, or 300 MBytes.
2. The private key size is much more compact, and is dominated by three matrices with 2^{14} bits each (hence 2^{13} bytes in total). If the matrices are deterministically produced out of some secret key (e.g., 128-bit), the description is even smaller.
3. The inverse polynomial of our schemes has degree $(t + 1)^2$ for S-boxes of size $2t + 1$.

The S-box size (length of χ) has negligible effect on the performance because of the internal structure of χ and its inverse (see Appendix). Hence it only affects the security of the scheme. We choose a single S-box $\mathbf{a}$ of length 127, so that its inverse has degree 64, and will show later that a system with small S-boxes is insecure.

In order to defeat decomposition algorithms and hide the ASASA structure we suggest using perturbation polynomials. More precisely, we propose 24 random polynomials of degree 4 for the perturbation layer $\mathbf{a}_p$. We pad each plaintext with 8 zero bits, so that for each guess the probability to fit the padding is 2^{-8}. As a result, we get 2^{16} candidate plaintexts, and then check if we correctly computed the noise. This filters out all wrong plaintexts with probability $1 - 2^{-8}$.

Overall security. We have found a number of attacks on the χ-scheme in different variants (Section 3), so it appears that its algebraic structure yields it vulnerable. Nevertheless, the variant with added perturbation remains unbroken, and we offer it as cryptanalytic challenge, but not for the practical use. We expect the perturbation theory to develop in the near future, which would suggest a more secure set of parameters.

2.5 Scheme with expanding S-boxes

This variant provides a more compact description of the scheme since we may switch to a larger field. First, we want the nonlinear layer be a degree-2 polynomial over $\mathbb{F}_q, q > 2$, and define the linear (affine) transformations over the same field. Though a few examples of bijective transformations of degree 2 over a field not equal to $\mathbb{F}_2$ exist [55], they appear to be vulnerable to Groebner basis attacks in our own experiments. As a solution, we suggest expanding S-boxes, whose output is twice as big as the input. It is rather easy to design *injective* S-boxes of degree 2 with this property. Indeed, a random function with expansion rate 2 has no collisions with probability around 1/2, and hence there are enough injective transformations of the desired form.

Here is the summary of the scheme:

– Input length 128 bits (32 variables), output length 512 bits (128 variables);
– All polynomials and affine transformations are defined over $\mathbb{F}_{16}$;
– S-boxes map 16 bits to 32 bits and hence are described by 8 degree-2 polynomials over $\mathbb{F}_{16}$ of four variables. The inverse is computed with lookup tables of size 2^{16}.
– The first nonlinear layer has 8 S-boxes and doubles the state size to 256 bits. The second layer has 16 S-boxes and further doubles to 512 bits. Accordingly, the affine transformations S, T, U operate on 128-, 256-, and 512-bit states, respectively.

The output of the scheme is a set of 2^7 degree-4 polynomials over $\mathbb{F}_{16}$ over 32 input variables (each variable is encoded with 4 bits). There are $\binom{2^5}{4} \approx 2^{16.5}$ possible terms, hence, taking 4-bit constants into account, each polynomial is described by $2^{20.5}$ bits, or $2^{20.5+7-3} = 2^{24.5}$ bytes, which is about 24 MBytes.

The private key is smaller: affine layers contain $2^{7+7+1} + 2^{6+6+1} + 2^{5+5+1} \approx 2^{14.2}$ elements of $\mathbb{F}_{16}$. The 48 S-boxes are described as $2^{5.5+3}$ polynomials of $21 \approx 2^{4.5}$ terms each, hence 2^{13} elements, plus a few noise polynomials. In total, the private key fits into 2^{14} bytes.

We also suggest using perturbation polynomials here. Due to the large expansion rate, we can use rather dense perturbation layer $\mathbf{a}_p$ and still ensure a unique decryption. We use two random polynomials over $\mathbb{F}_{16}$ of degree four at each S-box, hence 32 polynomials in total. While decrypting we face $16^2 = 2^8$ options for each S-box output. As a result, the probability of having non-unique decryption of the last S layer is $2^4 \cdot 2^{16+8-32} = 2^{-4}$, and if this happens the next layer filters out wrong candidates.

As we already mentioned, the expanding character of the scheme allows only public-key and white-box encryption, but not signature generation.

3 Security analysis of our schemes

In this section we apply various attacks to weakened versions of our schemes, thus demonstrating the design rationale behind them. We show that S-boxes in the χ-scheme must be large, that linearity (in contrast to affinity) of A may weaken the scheme, and that the expanding S-boxes should not be biased. Finally, we demonstrate that the added perturbations are crucial in both schemes, and that they must be secret. Our attacks are summarized in Table 1.

These attacks allow us to evaluate the security margin of the unbroken variants of our schemes. Since only the perturbation protects the χ-scheme from a number of practical attacks, we conclude that it is rather fragile, but might become a good candidate for a strong white-box implementation when the complexity of generic algorithms applied to the perturbed version is better understood. In contrast, the expanding scheme appears to be more resistant to generic attacks, and we propose it as a ready-to-use public-key encryption scheme and a strong white-box implementation.

Weakening	Attack complexity	Attack type	Reference
Expanding scheme			
Public perturbation	$2^{45} + D$	Interpolation	Section 3.2
Biased S-boxes (bias= 1/8)	2^{88}	LPN	Section 3.4
χ-scheme			
Public perturbation	$2^{57} + D$	Interpolation	Section 3.2
No perturbation	$\approx 2^{40}$	Groebner-basis	Section 3.3
Small S-boxes	2^{45}	Algebraic	Appendix A
Black-box n-bit ASASA with m-bit S-boxes			
-	$2^{(n-m)m}$	Multiset	Section 4.2

Table 1. Summary of our attacks on the weakened versions of our schemes. D stands for the complexity of decomposition attacks.

3.1 Generic attacks

Given the public-key of a multivariate scheme, an attacker may directly try to solve the multivariate polynomial equations using a generic algorithm. If the public-key is a vector of m polynomials in n over $\mathbb{F}_q$, then a plaintext can always be found by exhaustive search in time $\mathcal{O}(q^n)$. The other main family of algorithms to solve systems of polynomial equations are Groebner-basis algorithms, such as Buchberger's algorithm [20] and all its derivatives [36, 37].

Without going into details (the interested reader is referred to a standard textbook such as [25]), given a system of polynomial equations $f_1 = \cdots = f_m = 0$ in $x_1, \ldots, x_n$, a *Groebner basis* of the ideal spanned by the f_i's is an equivalent system of equations with nice properties. If the system admits a single solution $(a_1, \ldots, a_n)$, then a Groebner basis is precisely the vector of polynomials: $x_1 - a_1, \ldots, x_n - a_n$. It follows that if a Groebner basis can be computed, then the system of equations can be solved.

Groebner basis algorithms work by performing *polynomial elimination*, *i.e.,* by trying to eliminate some terms by summing suitable multiples of other polynomials. The complexities of these algorithms are difficult to analyze [3, 4]. They are essentially exponential in the highest degree reached by the polynomials created and manipulated by the algorithms during their execution. On "generic" systems of n equations in n variables, this degree is typically n. However, in some special cases it can be lower. For instance, the first HFE Challenge could be broken because in HFE, for some ranges of parameters, this degree was roughly $\mathcal{O}(\log n)$.

3.2 Interpolation attack on the ASASA scheme with public perturbation polynomials

We stressed that the perturbation polynomials must be secret. A reader may wonder why this is required, since these polynomials are seemingly mixed by the last affine transformation $\mathcal{U}$.

In this subsection we outline an attack that peels off the perturbation polynomials and recovers the core ASASA scheme in almost practical time. Suppose we work over a field $\mathbb{F}_2$ and the scheme adds perturbation polynomials at r bit positions after the nonlinear transformation $\mathbf{a}_2$ (cf. Eq. (2) and Figure 2), and the total number of variables in the scheme is n. Then we collect N plaintexts x_i such that

$$\mathbf{a}_p(x_i) = 0.$$

Since polynomials of $\mathbf{a}_p$ do not have any structure, finding a common zero is an NP-hard problem, and we expect that 2^r plaintexts must be tried to find a right one. Hence the naive complexity of this step is $N2^r$ evaluations[3] of $\mathbf{a}_p$.

Then we evaluate the right plaintexts on the perturbed scheme $\mathbf{b}_p$. Since $\mathbf{a}_p$ is zero, we have

$$\mathbf{b}_p(x_i) = \mathcal{U} \circ \mathbf{a}_2 \circ \mathcal{T} \circ \mathbf{a}_1 \circ \mathcal{S}(x_i).$$

Therefore, we know the evaluation of the ASASA scheme without perturbations on N plaintexts. Since the scheme has degree 4, the polynomial coefficients can be recovered by the Lagrange interpolation. There are $\sum_{i=0}^{4} \binom{n}{i}$ monomials of degree 4 or smaller, hence N must slightly exceed $\sum_{i=0}^{4} \binom{n}{i}$ to allow for linear dependencies among plaintexts. For the typical value $n = 2^7$ we need about 2^{25} right plaintexts to fully recover the core ASASA polynomials and then launch the decomposition attack. However, the interpolation itself is not a trivial procedure, since we deal with a multivariate function. Only recently an algorithm with complexity quadratic in the number of monomials has been proposed [1]. Equipped with it, we recover a single polynomial in 2^{50} bit operations, and the entire $\mathbf{b}_p$ in 2^{57} operations. In turn, 2^{25} right plaintexts can be obtained for 16 noisy bits in 2^{41} evaluations of $\mathbf{a}_p$, and for 24 noisy bits – in 2^{49} evaluations, which is close to 2^{55} bit operations. Therefore, the total complexity of recovering (x) is about 2^{57} bit operations.

This attack clearly shows that the perturbation polynomials must not be public and should not have any structure that would allow the adversary to find their common zeros. We do not see how the attack can be applied to secret polynomials.

[3] Finding subsequent solutions might be easier, but this step is not a dominant in our attack complexity.

3.3 Algebraic attack on the plain χ-scheme

Although χ has been used successfully in the symmetric world, it turns out to be a complete disaster in a multivariate context. An ASA construction where $S = \chi$, with $n = 127$ variables over $\mathbb{F}_2$ is broken in a few seconds by a direct Groebner basis computation. A two-layer ASASA construction is not more secure, and can be broken in less than two hours using the implementation of the F4 algorithm of the MAGMA computer algebra system [18] (and 100Gbytes of RAM). This happens because a Groebner basis can be computed by manipulating polynomials of small, constant degree (typically 3 or 6).

We work within the polynomial ring $R = \mathbb{F}_2 [x_0, \dots, x_{n-1}]$, and we consider the ideal of R:

$$\mathcal{I} = \left\langle f_0, \dots, f_{n-1}, \quad {x_0}^2 - x_0, \dots, {x_{n-1}}^2 - x_{n-1} \right\rangle$$

where $f_i = x_i + x_{i+2} + x_{i+1}x_{i+2} + a_i$ (all indices are taken modulo n), and where the a_i are constants. Any solution (in the x_i's) making all the polynomials in $\mathcal{I}$ vanish simultaneously, is a solution of $\chi(x_1, \dots, x_n) = (a_0, \dots, a_{n-1})$. Such a solution always exists, and is unique.

We will show that there are many linear polynomials in this ideal, and that they can be "easily" discovered (by manipulating small-degree polynomials). Indeed:

$$x_{i+1} \cdot f_i - x_{i+2} \cdot \left({x_{i+1}}^2 - x_{i+1}\right) + f_{i-1} = (x_{i-1} + x_{i+1}) - (a_{i-1} + a_{i+1})$$

The expression on the left-hand side is a polynomial combination of elements of $\mathcal{I}$, therefore it belongs to $\mathcal{I}$. As a consequence, the linear polynomial on the right-hand side can be found inside $\mathcal{I}$ after performing a few steps of polynomial elimination on polynomials of degree less than 3.

After these n linear relations have been found, another few steps of polynomial elimination allows all the variables but one to disappear. This shows that a Groebner basis of the ideal $\mathcal{I}$ can be computed in polynomial time. Now, performing a (random) linear change of coordinate in $\mathcal{I}$, or replacing the generators of $\mathcal{I}$ by (random) linear combinations thereof does not change this fact. As a conclusion, the ASA construction, where S is the χ-function, falls victim to a direct algebraic attack, by running any Groebner basis algorithm on the equations defining the "white-box".

This reasoning extends to the ASASA construction where both non-linear layers are χ (however, this time the degree is 6). It is an open question of how much the added perturbation slows the Groebner-basis attacks (our implementation does not break the selected noise parameters in reasonable time).

3.4 Attack on the expanding scheme with biased S-boxes

If S-box output bits are biased, an attack exploiting this bias can be applied. The last affine transformation can be viewed as affine over $\mathbb{F}_2$, so the further analysis without loss of generality applies to any field of characteristic two.

We target a single biased bit b after the second layer of expanding S-boxes: the probability $\mathbb{P}[b = 1]$ of its equality to 1 is equal to $p \neq \frac{1}{2}$. If y is a ciphertext, then following previous notations, the biased bit is the b-th component of $U^{-1} \cdot y$. In other terms, if $\overline{u}$ denotes the b-th line of U^{-1}, then $\langle \overline{u}, \overline{y} \rangle = b$.

Now, assume we collect a large number (say N) of ciphertexts. We stack them vertically into a matrix C, which thus has N rows. Let us also assume that b is biased towards zero. Then we have the "noisy linear system":

$$\overline{u} \cdot C = \overline{e},$$

where $\overline{e}$ is a vector of i.i.d. random variables following the Bernoulli distribution with mean p. Recovering $\overline{u}$ is exactly an instance of the Learning Parity with Noise (LPN) problem.

The best known algorithms to solve LPN are variants of the BKW algorithm [15], whose complexity is of order $\mathcal{O}\left(2^{n/\log n}\right)$. The only actual implementation (along with algorithmic improvements) is described in [56], and some more tweaks are given in [6]. The actual complexities of these algorithms depend on the bias (their efficiency decreases when the bias gets closer to zero).

With $n = 512$ variables, and if $\mathbb{P}[b = 1] = 1/8$, then the implementation of [56] is said to require 2^{80} bits of memory (plus the time needed to sort this much memory 80 times). However, time-memory tradeoffs, plus algorithmic improvements, allow [6] to conclude that the same problem can be solved in 2^{59} bits memory and less than 2^{100} bits operations. If 2^{80} bits of memory are available, then the running-time could be decreased to 2^{88} bit operations.

This beats more naive approaches, such as, for instance, enumerating all the possible sparse possibilities for the first n components of $\overline{e}$, and solving the corresponding linear system for each trial. The above instance would require more than 2^{120} operations to be solved using the naive approach.

Of course, the attack has to be repeated for each row of U^{-1}, and possibly twice for each row (assuming that the targeted bit is biased towards zero, or towards one). Note that the above estimates are extremely pessimistic; in random expanding S-boxes of degree 2, the biases we observed experimentally are much lower than what was used above (we observed $\mathbb{P}[b = 1] \approx 0.49$).

After U is recovered, we can view the output of expanding S-boxes, and are likely to recover them by interpolation due to low degree.

4 Towards secure weak white-box cryptography: symmetric ASASA schemes

In this section we propose a symmetric cipher based on five layers of random invertible affine transformations and S-boxes (ASASA). Due to availability of vector instructions in modern processors such schemes can also have a very fast software implementation.

4.1 Design

We propose a symmetric cipher with a classical set of parameters, widely used in AES and other designs. It has a block of $n = 128$ bits with $m = 8$ bit S-boxes and a choice of key-sizes $128 - 256$ bits. Let us outline specific parameters for linear and nonlinear layers.

Affine layers. A key-dependent $n \times n$ affine transformation can be produced out of the master key K by any secure key derivation function H_K (for example a fast stream cipher, or a block cipher in the counter mode (Appendix B)), and checking that the resultant matrix is invertible, this can be done in $O(n^3)$ steps, and we also generate an n-bit constant [4]. Since the matrix is random, we expect its branch number [28] to be close to n and any deviation to be undetectable because of its secrecy and large block size. Note that for each affine layer a new matrix is generated.

Nonlinear layers. Typically, nonlinear layers of symmetric ciphers consist of several small S-boxes, which have a compact description [16, 28]. For the ASASA scheme we propose to use 32 randomly generated 8-bit invertible S-boxes, which are all different and key-dependent. We note that efficiency of generic attacks on the SASAS structure [13] increases if smaller S-boxes are used, and thus it may be interesting in the future to explore full block size non-linear layers, for which such attack would not work.

Large S-box alternatives. The choice for large block algorithmic S-boxes is surprisingly limited. Unless the S-boxes are themselves multi-layer permutations (e.g., fixed-key ciphers), a compact description is typically delivered in the algebraic form as a function over an appropriate finite field. The resulting *permutation polynomials* have become an active research topic in the recent years. The well known example is X^{2^k+1} over $\mathbb{F}_{2^n}$ (scheme C* [59]); the more recent and interesting include $\left(X^{2^k} + X + a\right)^{-l} + X$ over $\mathbb{F}_{2^n}$ by Zeng et al. [81] (derived from Helleseth-Zinoviev polynomials) and $\left(X^{p^k} - X + a\right)^{\frac{p^n+1}{2}} + X^{p^k} + X$ over $\mathbb{F}_{p^n}$ where p is odd [82]. Further nontrivial examples can be found in [14, 21, 54, 55, 57]. It thus can be an interesting second challenge to break the symmetric ASASA scheme with known block-wide non-linear layers. Note however that fixed S-boxes do not offer implementation advantage and thus we would keep S-boxes secret and randomly generated in the main variant of our scheme.

[4] Non-anonymous final version of this paper will link to implementations of our schemes and challenges gradually increasing complexity for the interested cryptanalysts.

Implementation. The combined SA transformations can be implemented quite cheaply. It is well known from, e.g., fast AES implementations, that a multiplication of n-bit vector by $n \times n$-bit matrix over F_2 can be expressed as n XORs of n-bit vectors, and with table lookups — as $n/16$ XORs and $n/16$ lookups in tables of size 2^{16}. Let's consider the specific choice of $n = 128$ and $m = 8$ and assume that SIMD 128-bit operations can be performed in a single clock cycle (this is a reasonable assumption when a cipher is used, e.g., in the CTR mode). Then we need 16 table lookups and 16 XORs for the layers: A, SA, SA, which gives us 48/16=3 lookups and 3 XORs per byte, which is probably faster than optimized software implementations of AES [49] (without the special AES instructions). Total memory requirement for such implementation is: $3 \cdot 2^m(\frac{n}{m})^2 = 2^8 \cdot 16 \cdot 3 \cdot 16 \approx 196$ Kbytes. The private key size is even smaller: 14 KBytes (3 matrices of size 2^{11} bytes and 32 S-boxes of 2^8 bytes each). A byteslice implementation of the black-box ASASA in the CTR mode would require tables only 4 times as large as in AES, and would allow to work with a single table many times, so we do not expect cache problems on modern CPUs.

4.2 Security analysis

Differential and linear attacks. We expect the secret linear layers to hide all differential [10] and linear [58] properties of the cipher, since it becomes impossible to figure out any high-probability differential or linear trail. It can be argued, however, that the existence of high-probability characteristics may lead to efficient distinguishers. For instance, Dunkelman and Keller showed in [35] that if for every α the differential probabilities $\{\alpha \rightarrow \beta\}$ are much higher (or much lower) than for a random permutation, then this can be used as a distinguisher. The authors further suggested the parameter of effective linearity that essentially measures the average probability of the boomerang difference quartet (α, K) over all possible α, K and is supposed to take even unknown characteristics into account.

It is rather easy to show that only a tiny fraction of input differences may lead to high-probability differentials. Indeed, with probability $1-2^{(16-1)-64} = 1-2^{-49}$ at least 8 S-boxes at the first S layer are active. The highest differential probability is then delivered by a trail that activates all S-boxes in the second S layer. In turn, the highest differential probability that we expect for every input difference α can be upper bounded by 2^{-5}. Therefore, we expect that only 2^{-49} of all input differences yield characteristics with probabilities higher than $2^{-5 \cdot 24} = 2^{-120}$.

There exist characteristics that activate $16 + 1 = 17$ active S-boxes and probabilities of about 2^{-85} or slightly higher. They are even easy to find in the known-key setting. However, finding them in the secret-key setting is likely to require a birthday-like approach to activate only a few S-boxes. This implies the complexity close to 2^{60} to find such a characteristic, which puts the complexity of the attack close to the exhaustive search.

Finally, we have not found any use for the efficient linearity parameter, as it seems to be difficult to estimate it for our construction. The authors of [35] did it for the 2-round Feistel scheme surrounded by Vaudenay's decorrelation module [76], which should be compared to the SAS structure wrapped with A layers in our case. The latter problem is not only more difficult, but also would yield a random variable as the answer. Distinguishing whether this variable follows the distribution expected from the random permutation is a separate and challenging problem.

Other attacks. The boomerang and impossible differential attack can be also of concern. We have tried basic and improved versions of these attacks, and in all cases the randomness of the affine layers prevented us from mounting an attack. However, it is possible to build boomerang quartets in the known-key setting by activating a single S-box at both sides of the boomerang. Whether such properties can be carried out to the secret-key setting is the object of the future research. Impossible differential attacks typically rely on truncated differentials with probability 1 which exist in some ciphers due to incomplete diffusion. Since in our case the random affine layers provide complete diffusion and since the entrance into and the exit from the scheme are both guarded by these affine layers, chosen plaintext attacks have little chance of predicting truncated values somewhere inside the scheme.

Our scheme should be more secure than a two-round Even-Mansour cipher [17], where the subkeys are simply xored to the internal state (as opposed to applying a full-blown secret affine transformation to the internal state). The recent attack on the 2-round Even-Mansour [32] explicitly requires the access to the internal permutation and thus can not be immediately used in our setting.

The meet-in-the-middle attacks [29, 48] do not apply to our scheme, because the amount of key material used to compute any matching variable is too large (several S-boxes and a large part of the affine transformation). The cube attacks do not apply, since there is no compact polynomial representation of the scheme.

Structural attack. Finally, we investigate the structural attack from [13]. We will see that even though it does not apply to the 128-bit ASASA cipher, it allows to bound the security level of schemes with smaller block, which are used in Section 5. First, we recall the main property preserved by the SASA structure with m-bit S-boxes:

Theorem 1 ([13]). *Let $\{x_1, x_2, \ldots, x_{2^m}\}$ be the inputs to the SASA structure such that the input bits to one S-box take all possible combinations whereas the other bits are constant. Then the XOR of all outputs is the all-zero bit vector:*

$$\bigoplus_i = \mathbf{SASA}(x_i) = (0, 0, \ldots, 0).$$

Now consider the input y to some m-bit S-box in the first S layer of the ASASA scheme E. It is an affine function of the scheme input y:

$$y = M \cdot x \oplus c,$$

where M is an $(m \times n)$-matrix and c is a constant. Let L be an $(n \times n)$-matrix such that

$$M \times L = \big(M' \, 0 \, 0 \cdots \, 0,\big) \tag{3}$$

where M' is a $(m \times m)$- submatrix. If we apply E to $L \cdot x$, then y depends only on the first m bits of x. Thus we compose inputs $\{x_1, x_2, \ldots, x_{2^m}\}$ as in Theorem 1, multiply them by L and apply E. The output bits must sum to 0 bitwise. This property allows to recover the outer affine layer and eventually all the components of E. Equation (3) holds with probability $2^{-(n-m)m}$, which makes the attack impractical for large n. However, for small n it might be efficient. Therefore, the maximum security level of the ASASA scheme with n variables and m-bit S-boxes does not exceed $(n - m)m$ bits.

For our design, this gives the upper bound of 960 bits, which is far larger than the key length which is used to generate the affine and nonlinear layers. As a result, we claim the 128-bit security level for our design, even though a small factor over exhaustive key search might be saved by biclique attacks or by exploiting the full codebook. This is still higher security than what is offered by 2048 bit RSA.

5 Proposal for weak white-box security: ASASA-based block cipher

5.1 Weak white-box security

Definition 2. *Let the pair of algorithms (E, D) be a private-key encryption scheme, which takes key K as a parameter. We call $\mathfrak{F}(K)$ the* equivalent key set for key *K, if there is an algorithm*

$$\mathcal{A} : \mathcal{K} \to \mathcal{E}',$$

that produces an algorithm $\mathcal{E}'$ equivalent to E_K for any $\mathcal{K} \in \mathfrak{F}(K)$.

*We say that $\mathcal{O}_{E_K}$ is a T-*secure weak white-box implementation for *E_K if it is computationally hard to obtain $\mathcal{K} \in \mathfrak{F}(K)$ of length less than T given full access to $\mathcal{O}_{E_K}$.*

In other words, an adversary who gets a secure weak white-box implementation is unable to find out any compact (shorter than T) equivalent representation of it. In the practical sense, an adversary who wants to share a protected implementation of the encryption routine, would have to share the entire code. Such ciphers are motivated by DRM applications, which aim to prohibit the users of protected content from sharing the information needed to decrypt it. Clearly, in this context there is little practical difference between sharing the key and sharing, say, the set of subkeys as long as the other cipher operations are independent of the key. Therefore, naive methods of key protection, e.g. transforming it with a preimage-resistant hash function, would not prevent an attack. Ideally, the adversary would have to isolate and extract the entire decryption routine, which might be hard per se.

5.2 Weak white-box cipher proposal

In this section we propose a blockcipher family, which conforms to the weak white-box security notion, so that it is computationally infeasible to derive a key or any other compact secret information from the white-box implementation.

We further say that the white-box implementation is *memory-hard* if it requires a pre-specified and large enough amount of memory in the spirit of memory-hard key-derivation functions [70]. This concept is even stronger than the T-secure weak WB implementations, as an adversary is unable to reduce the implementation size at all and thus would have to publish the entire set of lookup tables. In contrast to earlier white-box designs, we offer a set of ciphers with a wide range of memory requirements.

Our memory-hard cipher consists of a number of smaller components, which are exposed as lookup tables in the white-box implementation. Each component is either a small-block ASASA cipher, adapted from the construction in Section 4, or just a single S-box. The S-boxes are minimum 8-bit wide to avoid equivalence problems [12], but 10-,and 12-bit ones are also used. All S-boxes and affine layers are derived in a deterministic way from the secret key. In fact, it is enough to have linear, not affine, layers, since the constant can be kept in the S-boxes. To estimate memory requirements, we assume for simplicity that each table output fits an integer number of bytes (e.g., 2 bytes for 10-bit S-boxes).

We propose the SPN structure for the cipher, i.e. we alternate layers of smaller ciphers (denote their number by R) with a public linear transformation $\mathcal{L}$. Any transformation with good diffusion shall be fine. For the 64-bit block it could be the MixBytes transformation of Grøstl [42], and for the 128-bit block — two pairs of this transformation separated by shuffling the 32-bit subwords between the outputs. Recalling that AES can be partitioned into 5 rounds with 4 32-bit Super S-boxes in each, we propose R layers for similar security margin. The cipher's pseudocode is as follows (Figure 3):

```
– Repeat $R$ times;
    • Apply $R$ parallel ASASA-based distinct blockciphers;
    • Apply the linear transformation $\mathcal{L}$ to the entire state.
```

We outline specific parameters and memory requirements for 64- and 128-bit blockciphers in Table 2. The 16-bit S layer has two 8-bit S-boxes, the 18-bit – 8-bit and 10-bit S-boxes, the 20-bit – two 10-bit S-boxes, and the 24-bit – three 8-bit S-boxes. We see that whereas the black-box implementation is a few dozen KBytes, the white-box implementation can be made large enough in the range from 2 MBytes to several GBytes.

Security analysis. Our ASASA components have very small block and only a few S-boxes in the S layer. Some attacks that are infeasible on the 128-bit block, may have practical complexity on the 16-bit block. The best attack we could find was presented in Section 4.2 and has complexity $2^{(n-m)m}$ for m-bit S-boxes and the n-bit block. As a result, the 16-bit ASASA components with 8-bit S-boxes have maximum security level of 64 bits, the 20-bit components — 100 bits, and 24-bit components with 8-bit S-boxes — 128 bits.

An easy way to increase the security level is to add two more layers, thus producing ASASASA components. This yields a 50% increase of the private key size, but no increase in the white-box

Rows	Component type	Components in row	Security level (bits)	White-box memory	Black-box memory
64-bit block					
4	ASASA	4×(16-bit)	64	2MB	16 KB
4	ASASA	3×(18-bit) + 10-bit	64	9 MB	32 KB
128-bit block					
8	ASASA	8×(16-bit)	64	8 MB	64 KB
8	ASASASA	8×(16-bit)	128	8 MB	96 KB
8	ASASA	6×(18-bit) + 2×(10-bit)	80	36 MB	130 KB
8	ASASA	24-bit + 6×(16-bit) +8-bit	64	384 MB	64 KB
8	ASASASA	24-bit + 6×(16-bit) +8-bit	128	384 MB	96 KB
6	ASASA	5×(24-bit) + 8-bit	128	1.4 GB	75 KB
5	ASASA	4×(28-bit) + (16-bit)	64	20 GB	85 KB
5	ASASASA	4×(28-bit) + (16-bit)	128	20 GB	130 KB

Table 2. Parameters and memory requirements of white-box and black-box implementations for the 128-bit blockcipher. We assume that n-bit component occupies $\lceil \frac{n}{8} \rceil 2^n$ bytes of memory in the white-box implementation.

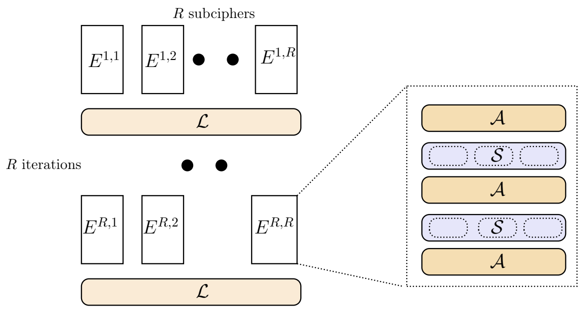

Fig. 3. Blockcipher family for weak white-box security.

implementation size. Since we have not found a way to expand our attack to this structure, we conjecture its security level to 128 bits. In Table 2 we provide both variants so that a protocol designer may choose between them according to his own requirements.

6 Conclusion

We have explored deeply the state of the art in black-box, white-box, and multivariate public key cryptography, and concluded that the ASASA structure is the minimal generic construction which is still unbroken. We constructed cipher candidates for all these settings. We designed two ASASA schemes for public-key cryptography based on multivariate polynomials. We showed how to avoid existing attacks on multivariate schemes, including the recent powerful decomposition algorithms by adding appropriate perturbation functions. In the traditional black-box setting we offered a cryptanalytic challenge of a fast cipher with small random S-boxes and random affine layers.

We proposed several solutions for white-box cryptography, both in weak and strong security notions. In the weak model, we designed a memory-hard cipher, which prohibits key extraction and requires an adversary to spend a large, pre-defined amount of memory. It is based on small ASASA components. We showed how our multivariate schemes can be used as strong white-box implementations, as they are not invertible without the key and allow fast encryption and decryption for legitimate users. We compare the implementation size of our schemes with other unbroken MQ-systems in Table 3.

Our findings indicate a number of future research directions. First, it would be interesting to explore algorithmic large S-boxes in the black-box ASASA structure, e.g. instantiated with recently found permutation polynomials. Secondly, a theory of perturbation layers as a countermeasure to generic decomposition algorithms needs to be developed, possibly along the concept of LWE (Learning with Error). Thirdly, we suggest investigating the actual security level of small-block (16-,20-, 24-bit) ASASA schemes to figure out which components are suitable for weak white-box implementations. Finally, open question is to develop constructions with smaller descriptions (e.g., within 1 MByte), which are bijective, suitable for digital signatures, and allow strong white-box implementations.

Scheme	Field	# variables	# polynomials	Degree	Private key size	PK/white-box size	Ref.
Black-box ASASA	$\mathbb{F}_2$	128	-	-	14 KB	196 KB	Sec. 4
χ-scheme*	$\mathbb{F}_2$	127	127	4	8 KB	300 MB	Sec. 2
Expanding scheme	$\mathbb{F}_{16}$	32	128	4	16 KB	24 MB	Sec. 2
Memory-hard cipher	$\mathbb{F}_2$	64	-	-	16-32 KB	2-8 MB	Sec. 5
Memory-hard cipher	$\mathbb{F}_2$	128	-	-	64-130 KB	8 MB – 20 GB	Sec. 5
HFE	$\mathbb{F}_{16}$	64	64	2	48 KB	520 KB	[65]
UOV	$\mathbb{F}_{256}$	78	26	2	71 KB	80 KB	[51]

* — several variants broken in this paper.

Table 3. Our schemes in comparison, along with (presumably) secure parameters for UOV and HFE.

References

1. Frederik Armknecht, Claude Carlet, Philippe Gaborit, Simon Künzli, Willi Meier, and Olivier Ruatta. Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. In *EUROCRYPT'06*, volume 4004 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2006.
2. Jean-Philippe Aumasson, Matthieu Finiasz, Willi Meier, and Serge Vaudenay. Tcho: A hardware-oriented trapdoor cipher. In *ACISP'07*, volume 4586 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2007.
3. Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université de Paris VI, 2004. In French.
4. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *Proc. International Conference on Polynomial System Solving (ICPSS)*, pages 71–75, 2004.
5. Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. Format-preserving encryption. In *Selected Areas in Cryptography'09*, volume 5867 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2009.
6. Daniel J. Bernstein and Tanja Lange. Never trust a bunny. In *RFIDSec*, volume 7739 of *Lecture Notes in Computer Science*, pages 137–148. Springer, 2012.
7. Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. RadioGatun, a belt-and-mill hash function, 2006. NIST Cryptographic Hash Workshop, available at `http://radiogatun.noekeon.org/`.
8. Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. The Keccak reference, version 3.0, 2011. `http://keccak.noekeon.org/Keccak-reference-3.0.pdf`.
9. Eli Biham. Cryptanalysis of Patarin's 2-round public key system with S-Boxes (2R). In *EUROCRYPT'00*, volume 1807 of *Lecture Notes in Computer Science*, pages 408–416. Springer, 2000.
10. Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer, 1993.
11. Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a white box AES implementation. In *Selected Areas in Cryptography'04*, volume 3357 of *Lecture Notes in Computer Science*, pages 227–240. Springer, 2004.
12. Alex Biryukov, Christophe De Cannière, An Braeken, and Bart Preneel. A toolbox for cryptanalysis: Linear and affine equivalence algorithms. In *EUROCRYPT'03*, volume 2656 of *Lecture Notes in Computer Science*, pages 33–50. Springer, 2003.
13. Alex Biryukov and Adi Shamir. Structural cryptanalysis of SASAS. In *EUROCRYPT'01*, volume 2045 of *Lecture Notes in Computer Science*, pages 394–405. Springer, 2001.
14. A. Blokhuis, R.S. Coulter, M. Henderson, and C.M. O-Keefe. Permutations amongst the Dembowski-Ostrom polynomials. *Finite Fields and Their Applications*, pages 37–42, 2001.
15. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.
16. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *CHES'07*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
17. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, François-Xavier Standaert, John P. Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations. In *EUROCRYPT'12*, volume 7237 of *Lecture Notes in Computer Science*, pages 45–62. Springer, 2012.

18. Wieb Bosma, John J. Cannon, and Catherine Playoust. The Magma Algebra System I: The User Language. *J. Symb. Comput.*, 24(3/4):235–265, 1997.
19. Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. White box cryptography: Another attempt. *IACR Cryptology ePrint Archive*, 2006. available at`http://eprint.iacr.org/2006/468`.
20. Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, University of Innsbruck, 1965.
21. Xiwang Cao and Lei Hu. New methods for generating permutation polynomials over finite fields. *Finite Fields and Their Applications*, 17(6):493–503, 2011.
22. Vincent Carlier, Hervé Chabanne, and Emmanuelle Dottax. Grey box implementation of block ciphers preserving the confidentiality of their design. *IACR Cryptology ePrint Archive*, 2004:188, 2004.
23. Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. White-box cryptography and an AES implementation. In *Selected Areas in Cryptography'02*, volume 2595 of *Lecture Notes in Computer Science*, pages 250–270. Springer, 2002.
24. Stanley Chow, Philip A. Eisen, Harold Johnson, and Paul C. van Oorschot. A white-box DES implementation for DRM applications. In *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2002.
25. David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, (Undergraduate Texts in Mathematics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1991.
26. Joan Daemen. *Cipher and Hash Function Design Strategies based on linear and differential cryptanalysis.* PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, March 1995.
27. Joan Daemen and Craig S. K. Clapp. Fast hashing and stream encryption with PANAMA. In *FSE'98*, volume 1372 of *Lecture Notes in Computer Science*, pages 60–74. Springer, 1998.
28. Joan Daemen and Vincent Rijmen. *The Design of Rijndael. AES — the Advanced Encryption Standard.* Springer, 2002.
29. Whitfield Diffie and Martin Hellman. Special feature exhaustive cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10:74–84, 1977.
30. Jintai Ding. A new variant of the Matsumoto-Imai cryptosystem through perturbation. In *Public Key Cryptography'04*, volume 2947 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 2004.
31. Jintai Ding and Dieter Schmidt. Cryptanalysis of HFEv and internal perturbation of HFE. In Serge Vaudenay, editor, *Public Key Cryptography'05*, volume 3386 of *Lecture Notes in Computer Science*, pages 288–301. Springer, 2005.
32. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key recovery attacks on 3-round Even-Mansour, 8-step LED-128, and full AES^2. In *ASIACRYPT'13*, volume 8269 of *Lecture Notes in Computer Science*, pages 337–356, 2013.
33. Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern. Practical cryptanalysis of SFLASH. In *CRYPTO*, volume 4622, pages 1–12. Springer, 2007.
34. Vivien Dubois, Louis Granboulan, and Jacques Stern. Cryptanalysis of HFE with internal perturbation. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography'07*, volume 4450 of *Lecture Notes in Computer Science*, pages 249–265. Springer, 2007.
35. Orr Dunkelman and Nathan Keller. A new criterion for nonlinearity of block ciphers. *IEEE Transactions on Information Theory*, 53(11):3944–3957, 2007.
36. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.
37. Jean-Charles Faugère. A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F5). In T. Mora, editor, *ISSAC '02: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pages 75–83, New York, NY, USA, July 2002. ACM Press. isbn: 1-58113-484-3.
38. Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *LNCS*, pages 44–60. Springer, 2003.
39. Jean-Charles Faugère and Ludovic Perret. Cryptanalysis of $2r^-$ schemes. In *CRYPTO'06*, volume 4117 of *Lecture Notes in Computer Science*, pages 357–372. Springer, 2006.
40. Jean-Charles Faugère and Ludovic Perret. An efficient algorithm for decomposing multivariate polynomials and its applications to cryptography. *J. Symb. Comput.*, 44(12):1676–1689, 2009.
41. Jean-Charles Faugère, Joachim von zur Gathen, and Ludovic Perret. Decomposition of generic multivariate polynomials. In *ISSAC'10*, pages 131–137. ACM, 2010.
42. Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schlffer, and Sren S. Thomsen. Grøstl – a SHA-3 candidate. Submission to NIST (Round 3), 2011.
43. Louis Goubin, Jean-Michel Masereel, and Michaël Quisquater. Cryptanalysis of white box DES implementations. In *Selected Areas in Cryptography'07*, volume 4876 of *Lecture Notes in Computer Science*, pages 278–295. Springer, 2007.
44. Louis Granboulan and Thomas Pornin. Perfect block ciphers with small blocks. In *FSE'07*, volume 4593 of *Lecture Notes in Computer Science*, pages 452–465. Springer, 2007.
45. Mathias Herrmann and Gregor Leander. A practical key recovery attack on basic TCHo. In *Public Key Cryptography'09*, volume 5443 of *Lecture Notes in Computer Science*, pages 411–424. Springer, 2009.

46. Viet Tung Hoang, Ben Morris, and Phillip Rogaway. An enciphering scheme based on a card shuffle. In *CRYPTO'12*, volume 7417 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2012.
47. Yun-Ju Huang, Feng-Hao Liu, and Bo-Yin Yang. Public-key cryptography from new multivariate quadratic assumptions. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 190–205. Springer, 2012.
48. Takanori Isobe. A single-key attack on the full GOST block cipher. *J. Cryptology*, 26(1):172–189, 2013.
49. Emilia Käsper and Peter Schwabe. Faster and timing-attack resistant AES-GCM. In *CHES'09*, volume 5747 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2009.
50. Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography.* Chapman and Hall/CRC Press, 2007.
51. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In *EUROCRYPT*, pages 206–222, 1999.
52. Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.
53. Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *CRYPTO'10*, volume 6223 of *Lecture Notes in Computer Science*, pages 631–648. Springer, 2010.
54. Gohar M. M. Kyureghyan. Constructing permutations of finite fields via linear translators. *J. Comb. Theory, Ser. A*, 118(3):1052–1061, 2011.
55. Yann Laigle-Chapuy. A note on a class of quadratic permutations over f_{2^n}. In *AAECC*, volume 4851 of *Lecture Notes in Computer Science*, pages 130–137. Springer, 2007.
56. Éric Levieil and Pierre-Alain Fouque. An improved lpn algorithm. In *SCN'06*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006.
57. Nian Li, Tor Helleseth, and Xiaohu Tang. Further results on a class of permutation polynomials over finite fields. *Finite Fields and Their Applications*, 22:16–23, 2013.
58. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.
59. Tsutomu Matsumoto and Hideki Imai. Public quadratic polynominal-tuples for efficient signature-verification and message-encryption. In *EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 419–453. Springer, 1988.
60. Wil Michiels and Paul Gorissen. Mechanism for software tamper resistance: an application of white-box cryptography. In *Digital Rights Management Workshop*, pages 82–89. ACM, 2007.
61. Wil Michiels, Paul Gorissen, and Henk D. L. Hollmann. Cryptanalysis of a generic class of white-box implementations. In *Selected Areas in Cryptography'08*, volume 5381 of *Lecture Notes in Computer Science*, pages 414–428. Springer, 2008.
62. Yoni De Mulder, Peter Roelse, and Bart Preneel. Cryptanalysis of the xiao - lai white-box AES implementation. In *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2012.
63. National Institute of Standards and Technology (NIST), available at `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`. *FIPS-197: Advanced Encryption Standard*, November 2001.
64. Jacques Patarin. Cryptoanalysis of the matsumoto and imai public key scheme of eurocrypt'88. In *CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 248–261. Springer, 1995.
65. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 1996.
66. Jacques Patarin, Nicolas Courtois, and Louis Goubin. FLASH, a fast multivariate signature algorithm. In David Naccache, editor, *CT-RSA*, volume 2020 of *Lecture Notes in Computer Science*, pages 298–307. Springer, 2001.
67. Jacques Patarin and Louis Goubin. Asymmetric cryptography with s-boxes. In *ICICS'97*, volume 1334 of *Lecture Notes in Computer Science*, pages 369–380. Springer, 1997.
68. Jacques Patarin, Louis Goubin, and Nicolas Courtois. $c^*_{\pm}$ and hm: Variations around two schemes of T. Matsumoto and H. Imai. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT*, volume 1514 of *Lecture Notes in Computer Science*, pages 35–49. Springer, 1998.
69. Kenneth G. Paterson. Imprimitive permutation groups and trapdoors in iterated block ciphers. In *FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 201–214. Springer, 1999.
70. Colin Percival. Stronger key derivation via sequential memory-hard functions. *Self-published*, 2009.
71. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
72. Vincent Rijmen and Bart Preneel. A family of trapdoor ciphers. In *FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 139–148. Springer, 1997.
73. Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 706–723. Springer, 2011.
74. Amitabh Saxena, Brecht Wyseur, and Bart Preneel. Towards security notions for white-box cryptography. In *ISC'09*, volume 5735 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009.
75. Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.
76. Serge Vaudenay. Decorrelation: A theory for block cipher security. *J. Cryptology*, 16(4):249–286, 2003.
77. Hongjun Wu, Feng Bao, Robert H. Deng, and Qin-Zhong Ye. Cryptanalysis of rijmen-preneel trapdoor ciphers. In *ASIACRYPT'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 126–132. Springer, 1998.
78. Brecht Wyseur. *White-Box Cryptography.* PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, March 2009.

79. Dingfeng Ye, Kwok-Yan Lam, and Zong-Duo Dai. Cryptanalysis of "2R" schemes. In *CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 315–325. Springer, 1999.
80. Adam L. Young and Moti Yung. Monkey: Black-box symmetric ciphers designed for monopolizing keys. In *FSE'98*, volume 1372 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 1998.
81. Xiangyong Zeng, Xishun Zhu, and Lei Hu. Two new permutation polynomials with the form $(x^{2^k}+x+d)^s+x$ over $\mathbb{F}_2^n$. *Appl. Algebra Eng. Commun. Comput.*, 21(2):145–150, 2010.
82. Zhengbang Zha and Lei Hu. Two classes of permutation polynomials over finite fields. *Finite Fields and Their Applications*, 18(4):781–790, 2012.

A Attack on χ-scheme with small S-boxes and linear transformations

Let's consider a weakened version of the χ-scheme, where we remove constants from the affine transformations, and consider small S-boxes. In this section we show a very efficient attack on the modified scheme, which works for very large block sizes.

The public key is then defined as follows:

$$\mathbf{b} = U \circ \mathbf{a} \circ T \circ \mathbf{a} \circ S, \tag{4}$$

where $\mathbf{a}$ consists of 14 9-bit S-boxes. Hence the inverse of b has degree $5^2 = 25$.

Let us exploit dependencies between the linear parts of $\mathbf{b}$ and $\mathbf{a}$:

$$\mathbf{b}^{(1)} = U \times \mathbf{a}^{(1)} \times T \times \mathbf{a}^{(1)} \times S. \tag{5}$$

Whereas S, T, U are regular matrices, the matrix $\mathbf{a}^{(1)}$ is singular, and its kernel is the linear span of the kernels of $\chi^{(1)}$. The latter consists of two vectors: $\mathbf{1} = (\underbrace{1, 1, \ldots, 1}_{9})$ and $\mathbf{0} = (0, 0, \ldots, 0)$, so has dimension 1, and

$$\dim \ker \mathbf{a}^{(1)} = 14.$$

Now we note that

$$\ker \mathbf{b}^{(1)} = \ker \left(\mathbf{a}^{(1)} \times T \times \mathbf{a}^{(1)} \times S\right)$$

so $\ker \mathbf{b}^{(1)} \leq 2 \cdot \ker \mathbf{a}^{(1)} = 28$. In fact, we expect that $\ker \mathbf{b}^{(1)} = 28$.

The attack proceeds as follows. We select vector $q \in \ker \mathbf{a}^{(1)} = (\underbrace{1, 1, \ldots, 1}_{63}, 0, 0, \ldots, 0)$. Then we randomly select vector $p \in \ker \mathbf{b}^{(1)}$. Since $S^{-1}(q) \in \ker \mathbf{b}^{(1)}$, we get

$$\mathbb{P}[S(p) = q] \geq 2^{-28}.$$

Assuming that $S(p) = q$, we compute $\mathbf{b}_p = \mathbf{b}(x + p)$:

$$\mathbf{b}_p = U \circ \mathbf{a} \circ T \circ \mathbf{a} \circ S(x + p) = U \circ \mathbf{a} \circ T \circ \mathbf{a}\,(S(x) + q) = U \circ \mathbf{a} \circ T \circ [\mathbf{a} + L_q] \circ S(x),$$

where $L_q(y) = \mathbf{a}(y) + \mathbf{a}(y + q)$ — a first-order derivative of $\mathbf{a}$ (hence a linear function of y). Due to specific form of L_q (see details below), we obtain the following matrix equation

$$\begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} = \mathbf{b}_p^{(1)} + \mathbf{b}^{(1)} = V \times L_q \times S = \begin{pmatrix} V_1 S_1' & V_1 S_2' \\ V_3 S_1' & V_3 S_2' \end{pmatrix}, \tag{6}$$

where $V = U \times \mathbf{a}^{(1)} \times T$.

Our next assumption is that both V_1 and V_3 are invertible, which for random V holds with the total probability 2^{-4} (Appendix C). Then we get a system of four matrix equations:

$$\begin{cases} A_1 = V_1 S_1'; \\ A_2 = V_1 S_2'; \\ A_3 = V_3 S_1'; \\ A_4 = V_3 S_2'. \end{cases} \tag{7}$$

where we multiply both sides by either $V_1^{(-1)}$ or $V_3^{(-1)}$. Hence we get a quadratic system of $4 \cdot 63^2$ linear equations, which we solve with probability 2^{-2}.

Therefore, we compute V_1 and V_3 from one guess of p with the total probability $2^{-28-4-2} = 2^{-34}$. Now we show how to exploit equivalences in linear transformations, increase the probability to 2^{-18}, and recover the whole V with complexity of about 2^{45} bit operations.

Details. We note that

$$\chi(x + \mathbf{1}) = \chi(x) + \overrightarrow{x} + \overrightarrow{\overrightarrow{x}},$$

where $\overrightarrow{x}$ is the rotation of x to the right by 1. Hence the matrix L_q is quasi-diagonal, with 7 matrices R_9 along the lead diagonal in the beginning and zeros afterwards.

$$L_q = \begin{bmatrix} \boxed{R_9} & & & & & \\ & \boxed{R_9} & & & 0 & \\ & & \ddots & & & \\ & & & \boxed{R_9} & & \\ & & & & 0 & \\ & 0 & & & & \ddots \\ & & & & & 0 \end{bmatrix}^{126 \times 126} \qquad R_9 = \begin{bmatrix} 0\,1\,1\,0 & & \\ & 0\,1\,1\,0 & \\ & \bullet\bullet\bullet & \\ 1 & & 0\,0\,0\,1 \\ 1\,1 & & 0\,0\,0 \end{bmatrix}^{9 \times 9}$$

Note that the position of blocks R_9 depend on the structure of q.

The linear part of $\mathbf{b}_p$ is defined as follows:

$$\mathbf{b}_p^{(1)} = U \circ \mathbf{a}^{(1)} \circ T \circ \left[\mathbf{a}^{(1)} + L_q\right] \circ S. \tag{8}$$

Summing Equations (5) and (8), we get

$$\mathbf{b}_p^{(1)} + \mathbf{b}^{(1)} = U \circ \mathbf{a}^{(1)} \circ T \circ L_q \circ S \tag{9}$$

Let us denote $U \circ \mathbf{a}^{(1)} \circ T$ by $V = \begin{pmatrix} V_1 & V_2 \\ V_3 & V_4 \end{pmatrix}$, where V_i are submatrices of size 63×63. Let also $S = \begin{pmatrix} S_1 & S_2 \\ S_3 & S_4 \end{pmatrix}$ with quadratic submatrices of the same size. Due to the form of L_q, we have that

$$L_q \times S = \begin{pmatrix} S_1' & S_2' \\ 0 & 0 \end{pmatrix}$$

and obtain Equation (6).

Improvement. First, we note that there are many equivalent ($\equiv$) transformation pairs (S, T). By the equivalence we understand that they produce the same $\mathbf{b}$. Indeed, since $\mathbf{a}$ is composed of identical S-boxes,

$$(S_1, T_1) \equiv (\pi^{-1} \circ S_1, T_1 \circ \pi),$$

where π is an arbitrary permutation of S-boxes.

Hence if $S(p) = \pi(q)$ for one of such permutations π, we still derive some meaningful information about V. Let Q_π be the matrix over $\mathbb{F}_2^{126 \times 126}$ corresponding to the permutation π of S-boxes. Then

$$\mathbf{b}_p = U \circ \mathbf{a} \circ T \circ \mathbf{a} \circ S(x + p) = U \circ \mathbf{a} \circ T \circ \mathbf{a}\,(S(x) + \pi(q)) = U \circ \mathbf{a} \circ T \circ \mathbf{a}\left(\pi\left(\pi^{(-1)}(S(x)) + q\right)\right) =$$
$$= U \circ \mathbf{a} \circ T \circ \pi \circ \mathbf{a}\left(\pi^{(-1)}(S(x)) + q\right) = U \circ \mathbf{a} \circ T \circ \pi \circ [\mathbf{a} + L_q] \circ \pi^{-1}(S(x))$$

Then we note that

$$\mathbf{a} \equiv \pi \circ \mathbf{a} \circ \pi^{-1} \implies Q_\pi \times \mathbf{a}^{-1} \times Q_\pi^{-1} = \mathbf{a}^{-1}$$

and recover the left half of $V \times Q_\pi$ instead of V.

Now we note that the left multiplication by Q_π permutes rows, whereas the right multiplication permutes columns. Hence for each guess of p we recover the left half of $V \times Q_\pi$ for some π, which yields some 7 out of 14 columns of V. Over the multiple guesses of p, the same columns get proposed multiple times. The attack algorithm then is as follows:

1. Fix $q \in \ker \mathbf{a}^{(1)} = (\underbrace{1, 1, \ldots, 1}_{63}, 0, 0, \ldots, 0)$.
2. Randomly select $p \in \ker \mathbf{b}^{(1)}$.
3. Assuming that $S(p) = \pi(q)$ for some π, try to recover the left half of $V \times Q_\pi$. If succeeded, store the resulting 7 columns in the memory.
4. If any column is proposed more than one time, store it in another table. Until all the 14 columns are proposed twice, go to step 1.

Let us compute the running time of the modified attack algorithm. There are $\binom{14}{7} \approx 2^{12}$ possible values of $\pi(q)$, hence the assumption at Step 3 holds with probability $2^{-28+12} = 2^{-16}$. The resulting system has 2^{-6} chance to resolve into 7 columns of the actual V. To make each column be proposed twice, we have to succeed at Step 3 four times. Hence Steps 1-3 are executed 2^{24} times on average. Step 3 is the bottleneck, as it inverts four matrices of size 2^6 and solve a system of size 2^7. This yields about 2^{20} bit multiplications and slightly larger number of xors. In total, we expect to recover V (or its equivalent) with complexity of around 2^{45} binary operations.

Having computed V, we multiply it by L_q and easily recover S^{-1} and then S. It remains to recover T and U, and we expect this operation to be less costly than the recovery of S.

B Layer generation procedure

In this section we give an example of how the layers of the ASASA schemes can be generated. Assume that we have the AES encryption procedure E and a properly generated master key K; if such routines are unavailable, then generic methods like HKDF [53] can be used.

– To generate the t-bit S-box, we produce $t2^t$ bits of keystream by encrypting plaintexts of the form

$$\underbrace{0}_{56\,bits} || \underbrace{q}_{8\,bits} || \underbrace{i}_{64\,bits},$$

where q is the number of the S-box and i is the counter. Each t-bit string selects an entry for $S(i)$ out of the previously unassigned values.

– To generate an invertible $n \times n$ matrix, we produce the keystream by encrypting

$$\underbrace{0}_{55\,bits}\, 1|| \underbrace{q}_{8\,bits} || \underbrace{i}_{64\,bits},$$

where q is the number of the A layer and i is the counter. When an n-bit string is produced, we add it to the matrix as a row and check if the matrix has the maximal rank. If not, the string is discarded.

It is clear that the employed domain separation method ensures that the counter values used to produce S-boxes and matrices do not overlap.

C Invertible matrices over $\mathbb{F}_2$

The number of invertible matrices of size $n \times n$ over $\mathbb{F}_2$ is computed as follows:

$$G_n = (2^n - 1) \cdot (2^n - 2) \cdot (2^n - 4) \cdots (2^n - 2^{n-1}) = 2^{n^2} \cdot (1 - \frac{1}{2^n}) \cdot (1 - \frac{1}{2^{n-1}}) \cdots (1 - \frac{1}{2}).$$

It is known that

$$G_n / 2^{n^2} > 0.288788.$$

D The inverse of χ

A generic algorithm to compute the inverse of χ is as follows:

```
1. Given $x = (x_0, x_1, \ldots, x_{k-1})$ find $\chi^{-1}(x)$;
2. Let $y \leftarrow x$;
3. For $0 \le i < 3\lfloor \frac{k}{2} \rfloor$
     – $y_{(k-2)i} \leftarrow y_{(k-2)i} \oplus x_{(k-2)i+2} \cdot x_{(k-2)i+1}$.
4. Return $y$.
```

The algorithm makes $O(k)$ bit operations. In practice, it can be optimized with bitslicing technique.

For the 9-bit S-box that is based on the χ transformation, the inverse polynomials have the following structure:

$$\begin{aligned}
p_3(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) &= x_3 + (x_5 + (x_7 + (x_0 + x_2 * (x_1 + 1)) * (x_8 + 1)) * (x_6 + 1)) * (x_4 + 1) = \\
&= x_3 + (x_5 + (x_7 + (x_0 + x_2 * \overline{x_1}) * \overline{x_8})\overline{x_6})\overline{x_4} = \\
&= x_3 + (x_5 + (x_7 + x_0 + x_2x_1 + x_2 + x_0x_8 + x_2x_1x_8 + x_2x_8)\overline{x_6})\overline{x_4} = \\
&= x_3 + (x_5 + x_7 + x_0 + x_2x_1 + x_2 + x_0x_8 + x_2x_1x_8 + x_2x_8 + x_7x_6 + \\
&\quad + x_0x_6 + x_2x_1x_6 + x_2x_6 + x_0x_8x_6 + x_2x_1x_8x_6 + x_2x_8x_6)\overline{x_4}.
\end{aligned}$$