9-30-2015

Affine-Power S-Boxes over Galois Fields with Area-Optimized Logic Implementations

Christopher Wood
caw4567@mail.rit.edu

Marcin Lukowiak
Rochester Institute of Technology, mxleec@rit.edu

Stanislaw Radziszowski
spr@cs.rit.edu

Affine-Power S-Boxes over Galois Fields with Area-Optimized Logic Implementations

Christopher A. Wood[1,2], Stanisław P. Radziszowski[2], Marcin Lukowiak[3]

[1] Department of Computer Science, UC Irvine
[2] Department of Computer Science, Rochester Institute of Technology
[3] Department of Computer Engineering, Rochester Institute of Technology

Abstract. Cryptographic S-boxes are fundamental in key-iterated substitution permutation network (SPN) designs for block ciphers. As a natural way for realizing Shannon's confusion and diffusion properties in cryptographic primitives through nonlinear and linear behavior, respectively, SPN designs served as the basis for the Advanced Encryption Standard and a variety of other block ciphers. In this work we present a methodology for minimizing the logic resources for n-bit affine-power S-boxes over Galois fields based on measurable security properties and finding corresponding area-efficient combinational implementations in hardware. Motivated by the potential need for new and larger S-boxes, we use our methodology to find area-optimized circuits for 8- and 16-bit S-boxes. Our methodology is capable of finding good upper bounds on the number of XOR and AND gate equivalents needed for these circuits, which can be further optimized using modern CAD tools.

Keywords: S-box design and construction, 16-bit S-boxes, composite Galois fields, S-box combinational circuits

1 Introduction

In order to enable area-efficient hardware implementations of block ciphers based on the key-iterated substitution permutation network (SPN) design principle, such as the Advanced Encryption Standard (AES) [13], the logic resources for each component or operation in the algorithm must be reduced as much as feasible. Traditionally, research efforts to minimize such resources have focused on the S-box - the only nonlinear operation in the algorithm. Minimizing the area required for the AES S-box has been the subject of intense research because combinational implementations of this component often consume a majority of the total logic needed for the algorithm. In addition, with the prevalence of side-channel attacks such as DPA and CPA [22, 4], combinational designs are necessary for secure implementations that cannot be easily exploited by these attacks.

To aid the implementation of future cryptographic primitives that rely on S-boxes, we present a comprehensive methodology for constructing and implementing cryptographically significant S-boxes with the goal of low-area combinational logic defined in terms of the number of XOR and AND gate requirements. An affine-power S-box $S(x)$ is a composite function $S(x) = A(P(x))$ consisting of a highly nonlinear power mapping P over a binary Galois field and an affine transformation A. In developing our methodology we build upon the exhaustive mixed basis approach of Canright [6] and combinational logic minimization techniques of Boyar and Peralta [2] used for the AES S-box. Our methodology is composed of three steps: finding suitable affine-power S-box constructions, programmatically and exhaustively searching for implementation parameters (i.e. subfield decompositions and basis representations) that permit area-optimized circuits, and then efficiently mapping them into technology dependent resources using modern CAD tools.

We applied our methodology to find area-optimized 8- and 16-bit S-boxes over binary Galois fields. Using the affine-inverse construction leveraged by the AES S-box, we exhaustively searched for area-optimized S-boxes over $GF(2^8)$ defined by all 30 irreducible polynomials over $GF(2)$ of degree 8. Our search produced an 8-bit S-box with 103 XOR and 36 AND gates using the field polynomial $t(v) = v^8 + v^6 + v^5 + v^4 + v^2 + v + 1$, surpassing Canright's optimized S-box circuit for the AES, which uses a different field polynomial, by a single XOR gate *prior* to further logic optimization techniques. We also found new implementation parameters for the AES S-box that yield a reduction in a single XOR gate prior to the application of Boyar and Peralta's logic optimization techniques. In addition, for the 21 smallest irreducible polynomials of degree 16 over $GF(2)$, we found several 16-bit S-box constructions that have small area footprints. For example, we found a set of implementation parameters that permit an area-optimized circuit composed of 1238 XOR and 144 AND gates. Even smaller gate counts were achieved for other polynomials.

2 Related Work

S-boxes in key-iterated SPN algorithms are often constructed as an affine transformation composed of an inverse power mapping over some Galois field, as is the case for the AES. This particular power mapping has many desired cryptographic properties that, in practice, effectively render many known cryptanalysis attacks ineffective. Consequently, much

of the research on low-area implementations for the affine-power S-boxes has focused on minimizing the combinational logic required for the multiplicative inverse calculation over binary Galois fields.

Out of all known methods to compute the multiplicative inverse in $GF(2^n)$, the use of subfield decomposition has been the most effective and accepted technique for implementing low-area circuits. Using composite field arithmetic, the inverse can be computed using the Itoh-Tsujii inversion algorithm [17] or by direct decomposition to bitwise operations over $GF(2)$ [30]. Under the assumption that the elements in a particular field are represented using a normal basis, the number of multiplications required in the Itoh-Tsujii inversion algorithm still yields complex combinational logic for small fields. In comparison, direct decomposition to $GF(2)$ has yielded significantly smaller circuits [33, 34, 24, 6, 26, 27, 3, 2].

To date, the smallest AES S-box using composite field arithmetic and other combinational logic minimization techniques is due to Boyar and Peralta, who found an implementation that required only 83 XOR/XNOR and 32 AND gates [2]. This fell below the previous area record of 104 XOR and 36 AND gate count by Canright in 2005 [6], which was found by trying all mixed basis representations of the field $GF(((2^2)^2)^2)$ to reduce the cost of relevant arithmetic when computing the multiplicative inverse and then factoring all basis change matrices needed to map between $GF(2^8)$ and $GF(((2^2)^2)^2)$. Boyar and Peralta improved upon Canright's results by swapping his $GF((2^2)^2)$ inversion circuit for their own optimized version and then performing subsequent combinational logic minimization on the entire S-box circuit. Given the effectiveness of this two-phase approach, we model our methodology after both of them.

3 Quantified Security of S-Boxes

With the continued improvement of cryptanalytic attacks that include different forms of linear and differential cryptanalysis [23, 1] and algebraic analysis [8, 10], among many others [16, 18, 21], it is critically important that cryptographic S-boxes do not exhibit weaknesses that can be exploited by these attacks. For example, linear cryptanalysis of SPN-based block ciphers exploits the existence of some linear combinations of input and output bits in the substitution step that occur with high (or low) probability [23]. Therefore, highly nonlinear S-boxes are ideal to reduce the probability that such linear combinations can be found and effectively exploited. To determine if an S-box has this property, we may compute the nonlinearity $\mathcal{N}_l(S)$ of a particular S-box mapping $S : GF(2)^n \rightarrow GF(2)^m$

as:

$$\mathcal{N}_l(S) = \min_{c \in S_2^m} \{\mathcal{N}_l(c \cdot S)\} = \min_{c \in S_2^m} \{\mathcal{N}_l(c_0 f_0 \oplus c_1 f_1 \oplus \cdots \oplus c_{m-1} f_{m-1})\}, \quad (1)$$

where

$$\mathcal{N}_l(f) = 2^{n-1} - \frac{1}{2} \max_{u \in GF(2)^n} |W_f(u)| \quad (2)$$

is the nonlinearity of a Boolean function $f : GF(2)^n \to GF(2)$, f_i are the m coordinate functions of S for $i = 1, \ldots, m$, $\cdot$ is the inner product operator of two Boolean functions, and $W_f(u) = \sum_{x \in GF(2)^n} (-1)^{f(x) \oplus (u \cdot x)}$ is the Walsh transform of f with respect to the input function u [12]. Other cryptographic metrics of interest include: the maximum and minimum entries of the linear and difference distribution table [23, 1], δ-differential uniformity [28, 29], resiliency and correlation immunity [7], component algebraic immunity [7], XL and XLS algebraic immunity [25, 8], interpolation polynomial algebraic complexity [18, 13], and branch number [13].

All of these metrics can be computed within a reasonable amount of time for single n-bit S-boxes over fields $GF(2^n)$ when $n \leq 16$. For example, directly computing the differential uniformity can be done in $\mathcal{O}(2^{3n})$ time, which is feasible for a single 16-bit S-box. This complexity, however, prohibited such computations for all 16-bit S-boxes considered in this work. By computing these metrics for S-box candidates we may quantitatively compare the security of different constructions. In general, we seek to build S-boxes that have high nonlinearity, low differential uniformity, high resiliency, high algebraic immunity, and high algebraic complexity. We focus on these metrics when selecting possible S-box constructions in the first step of our methodology. In this step we find an affine transformation that can be composed of a suitable power mapping for the S-box. We describe this procedure in the following section.

4 Constructing Suitable S-Boxes

Our focus is on S-boxes built from power mappings over Galois fields, i.e. functions of the form $S(x) = x^d$ where $x \in GF(2^n)$ and $0 \leq d < 2^n$. While other possible constructions exist, such as those based on the well-defined cryptographic properties of Boolean functions, there are several limitations that make these difficult to use in practice. For instance, they typically do not have compact algebraic expressions, which implies that hardware and software implementations often use lookup tables for such mappings. This may be acceptable for small n-bit S-boxes (i.e. $n \leq 8$), but not for $n \geq 16$.

Power mappings of the form $S(x) = x^d$ over the field $GF(2^n)$ are typically classified by their exponents d. The known exponents of power mappings over $GF(2^n)$ for even n with substantially high nonlinearity and good differential uniformity properties are shown in Table 1.

Table 1: Cryptographically-significant power mappings.

Name	*Exponent* (d)	*Ref.*
Inverse	$-1 \equiv 2^n - 2$	[29]
Gold	$2^k + 1$, $\gcd\{k, n\} = 1$ for some $1 \leq k \leq 2^n - 1$	[9]
Kasami	$2^{2k} - 2^k + 1$, $\gcd\{k, n\} = 1$ for some $1 \leq k \leq n/2$	[9]
Dobertin	$2^{4k+3k+2k+k} - 1$ over $GF(2^n)$ with $n = 5k$	[9]
Niho	$2^m + 2^{m/2} - 1$ over $GF(2^n)$ with $n = 2m + 1$ and m even $2^m + 2^{(3m+1)/2} - 1$ over $GF(2^n)$ with $n = 2m + 1$ and m odd	[9]
Welch	$2^m + 3$ over $GF(2^n)$ with $n = 2m + 1$	[9]

Since these S-boxes are intended for block ciphers, it is natural to impose the additional requirement that they are bijective. By Fermat's Little Theorem it is easy to see that this only occurs when $\gcd\{d, 2^n - 1\} = 1$. Interestingly, with this restriction and the constraint $n = m = 16$, many of the possible values for d are discarded and only the inverse exponent remains. See [35] for a proof of this claim.

Although these power mapping exponents are very well studied in the literature, we did not settle with them as the only candidates. In fact, for 8-bit S-boxes, we exhaustively computed the nonlinearity and δ-differential uniformity of all power mappings over $GF(2^8)$ using the AES field polynomial to determine if there exists suitable candidates that could be studied further. We found that there are only 8 distinct power mapping exponents and inverses (d, d^{-1}) such that $\delta = 4$ and $\mathcal{N}_L = 112$, the optimal values for such power mappings: $(127, 253)$, $(191, 251)$, $(223, 247)$, $(239, 239)$, $(247, 223)$, $(251, 191)$, $(253, 127)$, $(254, 254)$. We did not perform similar computations for 16-bit S-boxes, simply because these computations are much more expensive. Also, since it is not necessary that an S-box be invertible (i.e. if the block cipher using the S-box is operated in CTR-mode), then we need to only find area-optimized circuits for either d or d^{-1}.

After finding a candidate S-box construction with reasonable nonlinearity and δ-differential uniformity, our next task was to modify the constructions to increase the algebraic complexity (note that the algebraic

```
Input: S(·), S^{-1}(·), GF(2^n), n, d
Output: M, c
done := False
repeat
    M := RandomMatrix(GF(2), n, n)
    c := RandomElement(GF(2^n))
    if det(M) ≠ 0 then
        valid := True
        ForwardPairs, InversePairs := [ ]
        for each x ∈ GF(2^n) do
            if x is not a fixed point then
                z := M x^d + c
                ForwardPairs := Append(ForwardPairs, (x, z))
                x' := (M^{-1}(z + c))^{d^{-1}}
                InversePairs := Append(InversePairs, (z, x'))
            end
            else
                valid := False
                break
            end
        end
        if valid = True then
            p(y) := Interpolate(ForwardPairs)
            p^{-1}(y) := Interpolate(InversePairs)
            if #p(y) > n and #p^{-1}(y) > n then
                return M, c
            end
        end
    end
until done = False
```

Algorithm 1: Probabilistic affine transformation search procedure, where $\#p(y)$ (resp. $\#p^{-1}(y)$) is the number of terms in the interpolation polynomial $p(y)$ (resp. $p^{-1}(y)$).

expression of an interpolation polynomial for an S-box defined solely by a power mapping consists of a single term). In order to avoid interpolation attacks, such expressions should have more terms (be more complex). Perhaps the most common technique for increasing the complexity is to compose an affine transformation of one such power mapping. Cui and Cao [11] proved that the algebraic complexity for any affine-power S-box over $GF(2^n)$ is bounded by $n+1$. Algorithm 1 presents a probabilistic procedure to search for an appropriate affine transformation for affine-power S-boxes, characterized by a matrix $\mathbf{M}$ and constant vector c. Using the same rationale for the affine transformation selection presented by Daemen and Rijmen in [14], this procedure searches for affine transformations

that have a "complex algebraic expression if combined with the inverse mapping" and, together with the inverse operation, have "no fixed points and no opposite fixed points." In this context, a fixed point or opposite fixed point occurs when there exists an element $x \in GF(2^n)$ such that $S(x) \oplus x = 0^n$ or $S(x) \oplus x = 1^n$. Since there are no known attacks that exploit the existence of fixed points, we opted to lift this constraint if the pair $\mathbf{M}$ and c provide more opportunities for logic optimization than pairs that do not yield any fixed points.

5 Searching for Area-Efficient Tower Field Constructions

There are a variety of isomorphic representations for the fields $GF(2^8)$ and $GF(2^{16})$. Using composite arithmetic to compute the multiplicative inverse requires arithmetic operations such as addition, multiplication, squaring, and scaling (i.e. multiplication by a constant) in the subfields. The complexity of such arithmetic heavily depends on the representation of elements in the subfields. Polynomial arithmetic is generally more computationally efficient with polynomials of a smaller degree. This can be shown by deriving the expressions for the arithmetic operations in these subfields. For example, given an element $\epsilon \in GF(((2^2)^2)^2)$ (where $r(x) = x^2 + x + \Pi$ defines $GF((2^2)^2)$) represented in a polynomial basis $[1, X]$ with subfield coefficients δ_1 and δ_2, ϵ^{-1} can be computed as

$$\epsilon^{-1} = \delta_1(\delta_2^2 + \delta_1\delta_2 + \delta_1^2\Pi)^{-1}x + (\delta_1 + \delta_2)(\delta_2^2 + \delta_1\delta_2 + \delta_1^2\Pi)^{-1}.$$

If ϵ is represented in a normal basis $[X, X^{16}]$, the expression becomes

$$\epsilon_1^{-1} = ((\delta_1\delta_2 + (\delta_1 + \delta_2)^2\Pi)^{-1}\delta_2)x^{16} + (\delta_1\delta_2 + (\delta_1 + \delta_2)^2\Pi)^{-1}\delta_1)x.$$

Deriving a general expression for inversion in $GF((2^2)^4)$ depends on numerous factors, including the coefficients of the polynomial $r(x)$ and the basis representation. Given the numerous possibilities, we omit such derivations here, but it should be intuitively clear that the higher-degree polynomials representing elements in $GF((2^2)^4)$ will lead to less compact expressions than the simple quadratic extension case in which we can always find an irreducible polynomial $r(x)$ with a unit x coefficient. Consequently, we focus on the tower fields $GF((((2^2)^2)^2)^2)$ and $GF(((2^2)^2)^2)$ for $GF(2^{16})$ and $GF(2^8)$, respectively. Using such isomorphic representations, the cost of all arithmetic operations with respect to the subfields using a polynomial and normal basis is given in Table 2.

Table 2: Cost of arithmetic in $GF((q^2)^2)$ with respect to subfield $GF(q^2)$ (A)ddition, (M)ultiplication, (Sq)uare, (I)nversion, (Sc)ale, and (SS)quare-scale operations for polynomial and normal basis representations.

Operation	*Polynomial Basis*	*Normal Basis*
Inverse	$3M + 2A + 1I + 1SS$	$3M + 2A + I + 1SS$
Add	$2A$	$2A$
Multiply	$3M + 4A + 1Sc$	$3M + 4A + 1Sc$
Square	$2Sq + 1Sc + 1A$	$3A + 2Sq + 1Sc$

To this end, let $t(v)$ be a degree 16 (8) irreducible polynomial over $GF(2)$ for $GF(2^{16})$ (analogously $GF(2^8)$), $s(y) = y^2 + \Psi y + \Lambda$, be the irreducible polynomial for $GF((((2^2)^2)^2)^2)$, $r(x) = x^2 + \Theta x + \Pi$ be the irreducible polynomial for $GF(((2^2)^2)^2)$, $q(w) = w^2 + \Omega w + \Sigma$ be the irreducible polynomial for $GF((2^2)^2)$, and finally $p(v) = v^2 + v + 1$ be the *only* irreducible polynomial for $GF(2^2)$. We enforce $\Psi = \Theta = \Omega = 1$ to simplify field arithmetic. Also, we denote by V, W, X, and Y roots of the polynomials for the fields $GF(2^2)$, $GF((2^2)^2)$, $GF(((2^2)^2)^2)$, and $GF((((2^2)^2)^2)^2)$, respectively, and refer to the forward and inverse basis change matrices needed to map elements from $GF(2^8)$ and $GF(2^{16})$ to their isomorphic tower field partners as $\mathbf{T}$ and $\mathbf{T}^{-1}$.

Each irreducible polynomial for the fields $GF(2^2)$,...,$GF((((2^2)^2)^2)^2)$ will have two distinct conjugate roots, which we denote as the sets $\{V, V^2\}$, $\{W, W^4\}$, $\{X, X^{16}\}$, and $\{Y, Y^{256}\}$. A polynomial basis for any field can be formed by selecting one of these roots as a basis element in conjunction with the identity element 1, e.g. $[1, V]$ or $[1, V^2]$ for $GF(2^2)$, whereas a normal basis requires that both roots are used. For each possible combination of basis elements we then programmatically determine the combinational complexity of subfield arithmetic needed to compute the inverse.

For each combination of basis elements we also perform several arithmetic and logic optimizations. For instance, as Satoh [34] mentions, it is possible to save on the number of gates required for a circuit if there exists two $GF((2^m)^2)$ multipliers that have a shared input. This is because both the polynomial and normal multipliers need to compute the sum of the two coefficients for the input elements, as shown in Figure 1. Therefore, every shared input factor will save one addition in the subfield. In addition, polynomial and normal multipliers for elements in $GF((2^2)^2)$ and $GF(((2^2)^2)^2)$ each have three subfield multipliers that will share a common factor, thus saving additional sub-subfield addition operations.

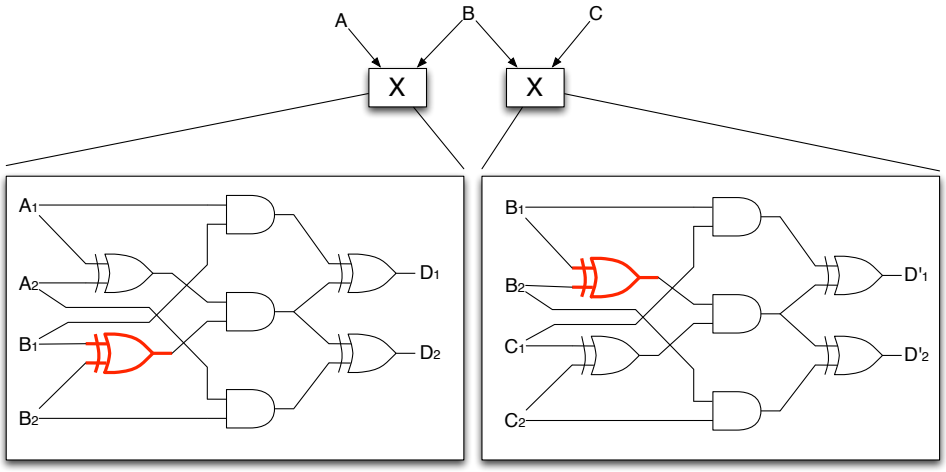

Fig. 1: XOR gate reduction for two $GF((2^2)^2)$ multipliers with a shared input B. The single XOR gate is saved by not recomputing the sum of the two B coefficients B_1 and B_2 of when B is represented in a normal basis.

We also make use of the optimizations to the square-scale operations performed by Canright [6]. At a high level, such optimizations are used to derive compact expressions for the square-scale operations given particular values of Π, which can only take a fixed number of values in order to make $r(x)$ irreducible over $GF((2^2)^2)$. We refer the reader to [35] and [6] for further discussion of these optimizations.

Our S-box construction program written in `Magma` [5] does not support exhaustive common subexpression elimination. This is primarily due to the fact that `Magma` does not support normal basis representations for finite field elements. Furthermore, exhaustively searching for all common subexpressions in all 432 possible inversion and square-scale algebraic expressions over $GF(((2^2)^2)^2)$ was outside the scope of this work. Future work will explore programmatically deriving such compact expressions in order to achieve lower gate counts. Also, it is important to note that, because we do not automatically apply the full set of Canright's optimizations, our gate counts will be *upper bounds* on the total number of gates. That is, the software that was written to count the number of gates for each field representation and basis selection will produce a result that is larger than or equal to what is presented in Canright's work, and as

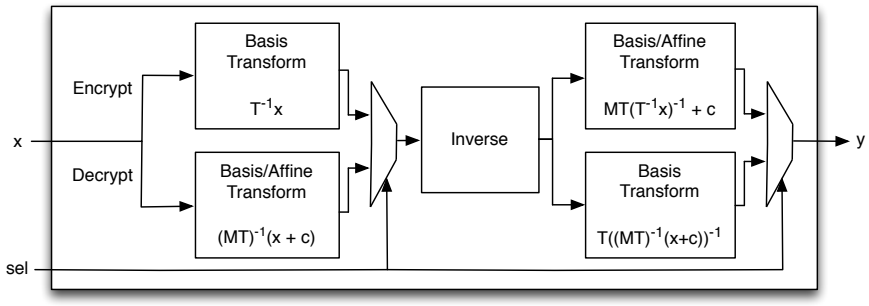

Fig. 2: High-level diagram for a merged S-box circuit. The sel signal is used to toggle encryption and decryption modes.

shown in his detailed report, other optimizations can be applied to lower this bound even further. After the tower field implementation parameters have been identified, we then utilize the logic minimization techniques of Paar [31] and Boyar and Peralta [3] to reduce the XOR gate count for the basis change matrices, which are merely linear mappings represented as straight-line programs (SLPs). An SLP for a binary matrix-vector multiplication expression is a finite sequence of lines of the form $u := \lambda v + \mu w$, where λ and μ are elements in $GF(2)$, u, v, and w are variables, and some lines are output of the corresponding multiplication.

In addition to these algebraic and gate-level optimizations, we also follow in the footsteps of Satoh [34] and Canright [6] by performing logic minimizations on merged S-box designs. The merged S-box design simply pairs the forward and inverse S-box operations into the same circuit that use the same inversion component, where the output is determined by a simple multiplexer. A high-level overview of the merged circuit is shown in Figure 2. We optimize the matrices $\mathbf{T}^{-1}/(\mathbf{MT})^{-1}$ and $\mathbf{MT}/\mathbf{T}$ separately.

6 New S-Box Constructions and Implementations

We measure the complexity, or cost, of a particular S-box as the total number of XOR and AND gates required in a combinational circuit implementation. To determine this cost for merged S-box circuits we measure the cost of the basis transformation matrices $\mathbf{T}$ and $\mathbf{T}^{-1}$ merged with the affine transformation matrices $\mathbf{M}$, the cost of a single inversion circuit, and the weight of the affine constant c. For fixed $\mathbf{M}$ and c, we perform an exhaustive search over all mixed basis representations of the S-box field

to find an upper bound on the gates required for the inversion circuit. We then use the logic optimization technique of Boyar and Peralta [2] to reduce the number of XOR gates required for the merged basis change and affine transformation matrices.

For 16-bit S-boxes, there are 128 choices for $s(y)$, eight choices for $r(x)$, two choices for $q(w)$, and only one choice for $p(v)$ that have a trace of unity. Since each of these polynomials has two distinct conjugate roots that can be used to represent the respective field elements with a polynomial or normal basis, there are exactly three basis element combinations in all degree-two extension subfields of $GF(2^{16})$. Consequently, there is a total of 165888 possible cases to consider for a single polynomial $t(v)$. Since the basis change matrices depend on the representation of $GF(2^{16})$, and there are 4080 candidates for $t(v)$, this means that we must consider about 6×10^8 possible cases to find a minimal transformation. Due to computational limitations, we selectively focused on the 21 smallest $t(v)$ polynomials when searching for 16-bit S-box implementation parameters. For 8-bit S-boxes, there are only 30 candidate $s(v)$ polynomials with smaller basis change matrices, so we did not have to impose a similar computational restriction.

We applied our methodology to find 8-bit S-boxes over $GF(2^8)$ and new 16-bit S-boxes over $GF(2^{16})$. For the 8-bit S-box case, we used Canright's optimized $GF(((2^2)^2)^2)$ inversion circuit when exhaustively searching for suitable implementation parameters. To perform this search, we consider all inverters which have a normal basis for $GF((2^4)^2)$ because the shared multiplication factor saves 5 XOR gates over inverters with a polynomial basis for $GF((2^4)^2)$. After Canright's optimizations, these S-boxes have anywhere from 66 to 68 XOR gates and 36 AND gates for the inverter [6]. Since the $GF(2^8)$ irreducible polynomial determines the number of XOR gates required for the basis change matrices $\mathbf{T}$ and $\mathbf{T}^{-1}$, we then considered all 30 degree 8 irreducible polynomials for $GF(2^8)$ to derive such basis change matrices. For each candidate inversion circuit and pair of basis change matrices $\mathbf{T}$ and $\mathbf{T}^{-1}$, we then applied the linear circuit minimization heuristic described by Boyar and Peralta in [3] to reduce the required XOR gates. This procedure was repeated for each irreducible polynomial $t(v)$ for $GF(2^8)$ and the basis representation that yielded the smallest number of required XOR and AND gates was recorded. Our results from this experiment for merged S-box designs are summarized in Table 2 in Appendix A.

We were able to improve upon Canright's S-box design using the AES polynomial by a single XOR gate, before logic gate optimizations such as using NAND/NOR instead of AND/XOR gates. With the same normal bases and coefficients Π and Σ, we found a different embedding of $GF(((2^2)^2)^2)$ into $GF(2^8)$ that yielded merged basis change and affine transformation matrices able to be implemented in only 37 XOR gates, as opposed to 38 found by Canright (see Figure 3 for the basis change matrices and corresponding SLP for proof). This single gate is saved in our field isomorphism and by applying Boyar and Peralta's optimization technique for the merged matrices $\mathbf{T}^{-1}/(\mathbf{MT})^{-1}$ and $\mathbf{MT}/\mathbf{T}$.

Out of all 30 degree 8 irreducible polynomials over $GF(2)$, we found that $t(v) = v^8 + v^6 + v^5 + v^4 + v^2 + v + 1$ permitted an S-box circuit with the smallest area requirement of only 103 XOR and 36 AND gates (see Table 3). Using this selection of $t(v)$, the basis change matrices to map an element $\alpha \in GF(2^8)$ represented in a polynomial basis to $\beta \in GF(((2^2)^2)^2)$ represented with the bases $[1, V]$, $[W, W^4]$, and $[X, X^{16}]$, where this tower field uses the coefficients $\Sigma = v$ and $\Pi = (v+1)w^4 + w$, require at most 35 XOR gates in the merged S-box design (see the SLP in Figure 4 for proof). Further area improvements for this S-box are likely possible by applying Boyar and Peralta's SLP minimization techniques, in addition to CAD-driven optimizations. However, even in its current state, this design surpasses Canright's optimized circuit for the AES S-box, and as such may be of value for implementations of future cryptographic algorithms.

We then considered the 21 smallest degree 16 irreducible polynomials $t(v)$ over $GF(2)$ in search for area-optimized 16-bit S-boxes. This search yielded several S-box constructions with small gate counts prior to (linear) logic optimizations of the basis change matrices. For the smallest irreducible polynomial $t(v) = v^{16} + v^5 + v^3 + v + 1$, we found a set of implementation parameters that permitted a circuit with a total of 1238 XOR and 144 AND gates. This candidate, shown in Figure 5, uses the basis sets $[1, V]$, $[1, W]$, $[1, X]$, $[Y^{256}, Y]$ to represent elements in $GF((((2^2)^2)^2)^2)$ and its respective subfields, where $\Sigma = v$, $\Pi = vw + v$, and $\Lambda = (vw + v)x + w$. The affine transformation and basis change matrices used to obtain the circuit are shown in Figure 5. A larger subset of these constructions are shown in Table 4 of Appendix A.

$$\mathbf{T}^{-1} = \begin{pmatrix} 1&1&1&0&0&1&1&1 \\ 0&1&1&1&0&0&0&1 \\ 0&1&1&0&0&0&1&1 \\ 1&1&1&0&0&0&0&1 \\ 1&0&0&1&1&0&1&1 \\ 0&0&0&0&0&0&0&1 \\ 0&1&1&0&0&0&0&1 \\ 0&1&0&0&1&1&1&1 \end{pmatrix} \qquad \mathbf{T} = \begin{pmatrix} 0&0&0&1&0&0&1&0 \\ 1&1&1&0&1&0&1&1 \\ 1&1&1&0&1&1&0&1 \\ 0&1&0&0&0&0&1&0 \\ 0&1&1&1&1&1&1&0 \\ 1&0&1&1&0&0&1&0 \\ 0&0&1&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0 \end{pmatrix}$$

Forward SLP for $\mathbf{T}^{-1}/(\mathbf{MT})^{-1}$		Inverse SLP for $\mathbf{MT}/\mathbf{T}$	
1) $y_5 = x_7$	19) $t_{18} = x_2 + t_{15}$	1) $y_{15} = x_5$	18) $t_{17} = x_0 + t_{12}$
2) $t_8 = x_1 + x_7$	20) $y_{13} = t_{18}$	2) $t_8 = x_2 + x_4$	19) $y_{13} = t_{17}$
3) $t_9 = x_2 + t_8$	21) $t_{19} = x_3 + t_9$	3) $y_0 = t_8$	20) $t_{18} = x_1 + x_6$
4) $y_6 = t_9$	22) $y_1 = t_{19}$	4) $t_9 = x_3 + x_6$	21) $y_{11} = t_{18}$
5) $t_{10} = x_6 + t_9$	23) $t_{20} = x_3 + t_{10}$	5) $y_8 = t_9$	22) $t_{19} = t_{16} + t_{18}$
6) $y_2 = t_{10}$	24) $y_{15} = t_{20}$	6) $t_{10} = x_1 + t_8$	23) $t_{20} = x_1 + x_7$
7) $t_{11} = x_0 + x_3$	25) $t_{21} = x_2 + t_{20}$	7) $t_{11} = x_5 + t_{10}$	24) $y_2 = t_{20}$
8) $y_8 = t_{11}$	26) $y_9 = t_{21}$	8) $t_{12} = x_2 + t_9$	25) $t_{21} = x_1 +_t 9$
9) $t_{12} = x_2 + t_{10}$	27) $t_{22} = x_5 + t_{13}$	9) $y_6 = t_{12}$	26) $y_7 = t_{21}$
10) $t_{13} = x_4 + t_{12}$	28) $y_7 = t_{22}$	10) $t_{13} = x_7 + t_{11}$	27) $t_{22} = x_2 + x_6$
11) $y_{11} = t_{13}$	29) $t_{23} = t_{10} + t_{15}$	11) $y_5 = t_{13}$	28) $y_{14} = t_{22}$
12) $t_{14} = x_1 + t_{11}$	30) $y_0 = t_{23}$	12) $t_{14} = x_0 + t_{13}$	29) $t_{23} = x_7 + t_{19}$
13) $y_{12} = t_14$	31) $t_{24} = t_{13} + t_{14}$	13) $y_{10} = t_{14}$	30) $y_9 = t_{23}$
14) $t_{15} = x_0 + x_5$	32) $y_4 = t_{24}$	14) $t_{15} = x_0 + x_4$	31) $t_{24} = t_9 + t_{11}$
15) $t_{16} = x_0 + t_9$	33) $t_{25} = x_0 + x_6$	15) $y_1 = t_{15}$	32) $y_{12} = t_{24}$
16) $y_3 = t_{16}$	34) $t_{26} = t_{24} + t_{25}$	16) $t_{16} = x_0 + t_8$	33) $t_{25} = t_9 + t_{19}$
17) $t_{17} = x_0 + t_{14}$	35) $y_{14} = t_{26}$	17) $y_3 = t_{16}$	34) $y_4 = t_{25}$
18) $y_{10} = t_{17}$			

Fig. 3: Forward SLP for $\mathbf{T}^{-1}/(\mathbf{MT})^{-1}$ and inverse SLP for $\mathbf{MT}/\mathbf{T}$ for use in Canright's design of the AES S-box [6]. Collectively, they require 37 XOR gates to implement.

$$z = S(x) = \begin{pmatrix} 0&0&0&0&1&1&0&1 \\ 1&1&0&0&1&0&1&1 \\ 0&0&1&0&0&1&1&1 \\ 0&1&0&0&1&1&1&1 \\ 1&1&0&0&0&0&0&0 \\ 1&0&0&1&1&0&0&0 \\ 0&0&0&0&1&1&0&0 \\ 1&0&0&0&1&0&1&0 \end{pmatrix} \begin{pmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad x = y^{-1} = (S^{-1}(z))^{-1} = \begin{pmatrix} 1&1&0&0&1&0&1&1 \\ 1&1&0&0&0&0&1&1 \\ 0&1&1&0&1&0&1&0 \\ 0&1&0&1&0&1&1&0 \\ 1&0&0&1&1&0&0&1 \\ 1&0&0&1&1&0&1&1 \\ 0&1&0&1&0&0&1&1 \\ 1&0&0&0&0&0&1&0 \end{pmatrix} \begin{pmatrix} z_7 \\ z_6 \\ z_5 \\ z_4 \\ z_3+1 \\ z_2 \\ z_1 \\ z_0 \end{pmatrix}$$

$$\mathbf{T} = \begin{pmatrix} 1&1&1&1&0&0&1&0 \\ 0&0&0&0&1&0&1&0 \\ 1&1&1&0&1&1&0&0 \\ 0&1&0&1&1&1&0&1 \\ 1&0&1&1&1&1&1&0 \\ 1&1&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&0 \\ 1&1&1&0&1&0&0&0 \end{pmatrix} \qquad \mathbf{T}^{-1} = \begin{pmatrix} 0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&1&1&1 \\ 1&0&1&0&1&0&1&0 \\ 1&1&0&0&0&0&0&1 \\ 1&0&0&0&1&1&1&0 \\ 0&0&1&0&0&0&0&1 \\ 1&1&0&0&1&1&1&0 \\ 0&1&0&1&1&0&0&1 \end{pmatrix}$$

Forward SLP for $\mathbf{T}^{-1}/(\mathbf{MT})^{-1}$		Inverse SLP for $\mathbf{MT}/\mathbf{T}$	
1) $y_0 = x_6$	17) $y_2 = t_{15}$	1) $y_2 = x_1$	19) $t_{18} = x_4 + t_{11}$
2) $y_9 = x_2$	18) $t_{16} = x_2 + x_7$	2) $y_{14} = x_0$	20) $y_7 = t_{18}$
3) $t_8 = x_0 + x_4$	19) $y_5 = t_{16}$	3) $t_8 = x_0 + x_1$	21) $t_{19} = x_2 + t_{18}$
4) $t_9 = x_6 + t_8$	20) $t_{17} = t_8 + t_{16}$	4) $t_9 = x_2 + x_4$	22) $y_{10} = t_{19}$
5) $y_{11} = t_9$	21) $t_{18} = x_1 + t_{11}$	5) $t_{10} = x_3 + x_6$	23) $t_{20} = x_3 + t_{12}$
6) $t_{10} = x_1 + x_7$	22) $y_6 = t_{18}$	6) $t_{11} = x_5 + t_8$	24) $y_4 = t_{20}$
7) $t_{11} = x_5 + t_9$	23) $t_{19} = x_3 + t_9$	7) $y_{13} = t_{11}$	25) $t_{21} = x_4 + x_6$
8) $y_4 = t_{11}$	24) $y_{12} = t_{19}$	8) $t_{12} = t_8 + t_9$	26) $y_9 = t_{21}$
9) $y_{15} = t_{11}$	25) $t_{20} = x_6 + t_{12}$	9) $y_{15} = t_{12}$	27) $t_{22} = t_{11} + t_{14}$
10) $t_{12} = x_3 + t_{10}$	26) $y_8 = t_{20}$	10) $t_{13} = x_0 + t_{10}$	28) $y_{12} = t_{22}$
11) $t_{13} = x_4 + t_{12}$	27) $t_{21} = t_{10} + t_{17}$	11) $y_0 = t_{13}$	29) $t_{23} = t_{11} + t_{16}$
12) $y_7 = t_{13}$	28) $y_{13} = t_{21}$	12) $t_{14} = t_{12} + t_{13}$	30) $y_1 = t_{23}$
13) $y_{14} = t_{13}$	29) $t_{22} = t_{11} + t_{17}$	13) $y_6 = t_{14}$	31) $t_{24} = t_{15} + t_{16}$
14) $t_{14} = x_0 + t_{10}$	30) $y_1 = t_{22}$	14) $t_{15} = x_3 + x_4$	32) $y_8 = t_{24}$
15) $y_3 = t_{14}$	31) $t_{23} = t_{14} + t_{15}$	15) $y_3 = t_{15}$	33) $t_{25} = x_0 + t_{17}$
16) $t_{15} = x_2 + t_9$	32) $y_{10} = t_{23}$	16) $t_{16} = x_6 + t_{12}$	34) $t_{26} = t_{18} + t_{25}$
		17) $t_{17} = x_3 + x_7$	35) $y_{11} = t_{26}$
		18) $y_5 = t_{17}$	

Fig. 4: The 8-bit S-box and basis change matrices for polynomial $t(v) = v^8+v^6+v^5+v^4+v^2+v+1$. The vector y is the inverse of the element x in $GF(2^8)$ (or $\bar{0}$ if $x = 0$). Accordingly, the output $y = S^{-1}(z)$ is inverted in the same way to obtain the original element x. The forward S-box SLP for $\mathbf{T}^{-1}/(\mathbf{MT})^{-1}$ and inverse S-box SLP for $\mathbf{MT}/\mathbf{T}$ are also shown, which collectively require 35 XOR gates to implement.

$$z = S(x) = \begin{pmatrix} 0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,1\,1\,0 \\ 1\,1\,0\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0 \\ 1\,1\,0\,0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,0\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0 \\ 1\,1\,0\,0\,0\,1\,1\,0\,0\,1\,1\,1\,1\,0\,1\,1 \\ 0\,1\,0\,0\,0\,0\,1\,1\,0\,1\,1\,1\,1\,1\,0\,1 \\ 0\,0\,1\,0\,1\,0\,1\,0\,1\,1\,0\,0\,1\,1\,0\,0 \\ 1\,0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0\,1\,1\,1 \\ 0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,0\,1 \\ 1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,1\,0\,1\,0\,0\,0 \\ 1\,0\,1\,0\,0\,1\,1\,1\,0\,0\,1\,1\,0\,1\,0\,0 \\ 1\,0\,1\,1\,1\,0\,1\,1\,1\,1\,0\,1\,1\,0\,0\,1 \\ 1\,0\,1\,0\,0\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,1 \\ 0\,1\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,0\,1\,0\,1\,1\,0\,1\,0\,0\,1\,1\,0\,0\,0 \end{pmatrix} \begin{pmatrix} y_{15} \\ y_{14} \\ y_{13} \\ y_{12} \\ y_{11} \\ y_{10} \\ y_9 \\ y_8 \\ y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad x = y^{-1} = (S^{-1}(z))^{-1} = \begin{pmatrix} 0\,1\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,0\,0\,0\,1 \\ 1\,1\,0\,1\,0\,0\,1\,0\,1\,0\,1\,1\,1\,1\,0\,1 \\ 1\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,1\,0\,1\,0\,1\,1\,1\,0\,1\,0 \\ 1\,1\,1\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1 \\ 1\,0\,1\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,1\,1 \\ 0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0 \\ 1\,1\,1\,0\,0\,1\,1\,1\,1\,0\,1\,1\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,1\,0\,1\,0\,1\,1 \\ 1\,0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,1 \\ 1\,0\,0\,0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,0 \\ 1\,0\,0\,0\,0\,1\,1\,1\,1\,1\,0\,1\,1\,1\,1\,1 \\ 1\,1\,1\,0\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,1 \end{pmatrix} \begin{pmatrix} x_{15} \\ x_{14}+1 \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10}+1 \\ x_9 \\ x_8+1 \\ x_7+1 \\ x_6 \\ x_5+1 \\ x_4+1 \\ x_3 \\ x_2+1 \\ x_1+1 \\ x_0+1 \end{pmatrix}$$

$$\mathbf{T} = \begin{pmatrix} 0\,1\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0 \\ 0\,1\,1\,0\,0\,1\,1\,1\,0\,0\,1\,0\,0\,1\,1\,1 \\ 0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \\ 1\,1\,0\,0\,0\,0\,1\,1\,0\,1\,0\,1\,0\,0\,1\,1 \\ 1\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,1 \\ 0\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,1\,0\,1\,0\,1\,0\,0\,1\,1\,1\,0\,1\,0\,0 \\ 1\,0\,1\,0\,0\,1\,1\,1\,1\,0\,1\,1\,0\,0\,1\,1 \\ 0\,0\,0\,1\,0\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,1 \\ 1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0\,1 \\ 0\,1\,1\,1\,0\,1\,0\,0\,1\,1\,1\,0\,0\,0\,1\,0 \\ 0\,1\,0\,1\,1\,1\,0\,0\,1\,1\,0\,1\,0\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1\,0\,0\,0\,0 \end{pmatrix} \qquad \mathbf{T}^{-1} = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,1\,0 \\ 1\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,0 \\ 1\,1\,0\,0\,1\,0\,0\,0\,1\,1\,0\,1\,1\,0\,1\,0 \\ 0\,1\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,1\,0\,1\,0\,0\,0\,0\,1\,1\,0 \\ 1\,0\,1\,1\,0\,0\,1\,1\,0\,0\,0\,1\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0\,1\,0 \\ 1\,0\,1\,0\,0\,0\,1\,1\,0\,1\,0\,1\,1\,0\,1\,0 \\ 1\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,1\,0\,1\,0\,1\,1\,1\,0\,1\,0 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,1\,1\,0\,1\,0\,0\,1\,1\,1\,0\,1\,1\,0 \\ 0\,1\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,0\,0 \\ 1\,1\,0\,1\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,1\,1 \end{pmatrix}$$

Fig. 5: The 16-bit S-box and basis change matrices for the polynomial $t(v) = v^{16} + v^5 + v^3 + v + 1$. The vector y is the inverse of the element x in $GF(2^{16})$ (or $\bar{0}$ if $x = 0$). Accordingly, the output $y = S^{-1}(z)$ is inverted in the same way to obtain the original element x.

7 Conclusion

In this work we presented a comprehensive methodology for identifying cryptographically significant S-box constructions based on power mappings over $GF(2^n)$ and searching for composite-field representations that permit low-area hardware implementations. We applied our technique to 8-bit S-boxes defined over $GF(2^8)$ using all 30 degree 8 irreducible polynomials and found several circuits with area-optimized implementations on par with or surpassing the AES equivalent (pending CAD optimizations). Motivated by a potential need for larger S-boxes, we then scaled up our procedure to 16-bit S-boxes. We believe this methodology and our results may be useful in the design of future cryptographic algorithms.

References

1. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-Like Cryptosystems. *Journal of Cryptology* **4.1** (1991), 3-72.
2. Joan Boyar and René Peralta. A Small Depth-16 Circuit for the AES S-Box. *IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg* 376 (2012), 287-298.
3. Joan Boyar, Philip Matthews, and René Peralta. Logic Minimization Techniques with Applications to Cryptology. *Journal of Cryptology* (2012), 1-33.
4. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation Power Analysis with a Leakage Model. *Cryptographic Hardware and Embedded Systems - CHES 2004, Springer Berlin Heidelberg* (2004), 16-29.
5. John Cannon and Allan Steel. The Magma computational algebra system. *Software available online* (magma.maths.usyd.edu.au) (2005).
6. David Canright. A Very Compact S-Box for AES. *CHES 2005 - Cryptographic Hardware and Embedded Systems, Springer Berlin Heidelberg* (2005), 441-455.
7. Claude Carlet and Emmanuel Prouff. On a New Notion of Nonlinearity Relevant to Multi-output Pseudo-random Generators. *Selected Areas in Cryptography, Springer Berlin Heidelberg* (2004).
8. Nicolas T. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. *Advances in Cryptology - ASIACRYPT 2002, Springer Berlin Heidelberg*, (2002), 267-287.
9. Nicolas T. Courtois, Blandine Debraize, and Eric Garrido. On Exact Algebraic [Non-]Immunity of S-Boxes Based on Power Functions. *Information Security and Privacy, Springer Berlin Heidelberg* (2006).
10. Gregory V. Bard, Nicolas T. Courtois, and Chris Jefferson. Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over $GF(2)$ via SAT-Solvers. *Presented at ECRYPT workshop Tools for Cryptanalysis* (2007).
11. Lingguo Cui and Yuanda Cao. A New S-Box Structure Named Affine-Power-Affine. *International Journal of Innovative Computing, Information and Control* 3.3 (2007), 751-759.
12. Thomas W. Cusick and Pantelimon Stănică. Cryptographic Boolean Functions and Applications. *Academic Press* (2009).

13. Joan Daemen and Vincent Rijmen. Advanced Encryption Standard (AES) (FIPS 197). *Technical report, Katholijke Universiteit Leuven/ESAT* (2001).
14. Joan Daemen and Vincent Rijmen. The Design of Rijndael: AES-the Advanced Encryption Standard. *Springer* (2002).
15. Hans Dobbertin. Almost Perfect Nonlinear Power Functions on $GF(2^n)$: The Welch Case. *IEEE Transactions on Information Theory* 45(4) (1999), 1271-1275.
16. Henri Gilbert and Thomas Peyrin. Super-Sbox Cryptanalysis: Improved Attacks for AES-like Permutations. *Fast Software Encryption, Springer Berlin Heidelberg* (2010).
17. Toshiya Itoh and Shigeo Tsujii. A Fast Algorithm for Computing Multiplicative Inverses in $GF(2^m)$ Using Normal Bases. *Information and Computation* 78.3 (1988), 171-177.
18. Thomas Jakobsen and Lars R. Knudsen. The Interpolation Attack on Block Ciphers. *4th International Workshop on Fast Software Encryption LNCS, Springer* 1267 (1997), pp. 28-40.
19. Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin. Improved Cryptanalysis of AES-like Permutations. *Journal of Cryptology* (2013), 1-27.
20. Alan Kaminsky, Michael Kurdziel, Stanisław Radziszowski. An Overview of Cryptanalysis Research of the Advanced Encryption Standard. *Proceedings of MILCOM'2010, San Jose, CA* (2010).
21. Lars R. Knudsen. Truncated and Higher Order Differentials. *Fast Software Encryption, Springer Berlin Heidelberg* 1008 (1995).
22. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. *Advances in Cryptology - CRYPTO99, Springer Berlin Heidelberg* (1999).
23. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. *Advances in Cryptology - EUROCRYPT93, Springer Berlin Heidelberg*, (1994).
24. Nele Mentens, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. A Systematic Evaluation of Compact Hardware Implementations for the Rijndael S-Box. *Topics in Cryptology-CT-RSA, Springer Berlin Heidelberg* (2005), 323-333.
25. Nawaz Yassir, Kishan Chand Gupta, and Guang Gong. Algebraic Immunity of S-Boxes Based on Power Mappings: Analysis and Construction. *IEEE Transactions on Information Theory* 55.9 (2009), 4263-4273.
26. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Using Normal Bases for Compact Hardware Implementations of the AES S-box. *Security and Cryptography for Networks. Springer Berlin Heidelberg* (2008), 236-245.
27. Yasuyuki Nogami, Kenta Nekado, Tetsumi Toyota, Naoto Hongo, and Yoshitaka Morikawa. Mixed Bases for Efficient Inversion in $F_{((2^2)^2)^2}$ and Conversion Matrices of SubBytes of AES. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E94-A:6 (2011), 1318-1327.
28. Kaisa Nyberg. Perfect nonlinear S-boxes. *Advances in Cryptology - EUROCRYPT91*. Springer Berlin Heidelberg (1991).
29. Kaisa Nyberg. Differentially Uniform Mappings for Cryptography. *Advances in Cryptology - Eurocrypt93*. Springer Berlin Heidelberg (1994).
30. Christof Paar. Some Remarks on Efficient Inversion in Finite Fields. *1995 IEEE International Symposium on Information Theory* (1995).
31. Christof Paar. Optimized Arithmetic for Reed-Solomon Encoders. *Proceedings of the 1997 IEEE International Symposium on Information Theory* (1997).
32. Vincent Rijmen. Efficient Implementation of the Rijndael S-box. *Katholieke Universiteit Leuven, Dept. ESAT, Belgium* (2000).

33. Atri Rudra, Pradeep K. Dubey, Charanjit S. Jutla, Vijay Kumar, Josyula R. Rao, and Pankaj Rohatgi. Efficient Rijndael Encryption Implementation with Composite Field Arithmetic. *Cryptographic Hardware and Embedded Systems - CHES, Springer Berlin Heidelberg* (2001).
34. Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A Compact Rijndael Hardware Architecture with S-Box Optimization. *Advances in Cryptology - ASIACRYPT, Springer Berlin Heidelberg* (2001), 239-254.
35. Christopher A. Wood. Large Substitution Boxes with Efficient Combinational Implementations. *M.S. Thesis, Computer Science, Rochester Institute of Technology* (August 2013).

A S-Box Constructions

In this appendix we provide detailed parameters for a variety of S-box constructions. A basis $B = [\beta^{n_1}, \beta^{n_2}]$ is used to represent an arbitrary element $\alpha \in GF((q^m)^2)$ as $\alpha = a_1\beta^{n_1} + a_2\beta^{n_1}$ for some $a_1, a_2 \in GF(q^m)$. The basis element powers n_1 and n_2 are chosen such that B is a polynomial or normal basis. In particular, if $n_1 = 0$, then B must be a polynomial basis where $n_2 \in \{1, q^m\}$. If $n_1 \neq 0$ and $n_2 \neq 0$ then B must be a normal basis. The order of normal basis elements in B depends on how `Magma` selects primitive elements. Specifically, if β^{q^m} is chosen as the primitive element then our S-box construction program will fix the basis $B = [\beta^{q^m}, \beta]$.

In the following tables we encode each irreducible polynomial $t(v)$, constant c, and binary matrix ($\mathbf{T}$, $\mathbf{T}^{-1}$, and $\mathbf{M}$ in row order), which are described in Sections 4 and 5, as a hexadecimal string. Σ, Π, and Λ, the irreducible polynomial coefficients described in Section 5, are shown with a polynomial basis. We also use the notation $\mathbb{F}!v$ to denote the embedding of v into the field $\mathbb{F}$, where $\mathbb{F} = GF(2^8)$ or $\mathbb{F} = GF(2^{16})$ for 8 and 16-bit S-boxes, respectively. The subfield bases are exactly as described in Section 5. Finally, the Inv. and Total fields denote the number of XOR gates required for the multiplicative inverse and merged S-box circuits, respectively.

Table 3: Area-optimized 8-bit S-box constructions for merged circuit implementations using all 30 irreducible polynomials $t(v)$ of degree 8 over $GF(2)$. The entry with polynomial $t(v) =$ 177 corresponds to the construction given in Figure 4.

$t(v)$	Σ	Π	$\mathbb{F}!v$	$\mathbb{F}!w$	$\mathbb{F}!x$	Bases	$\mathbf{T}$	$\mathbf{T}^{-1}$	$\mathbf{M}$	c	Inv.	Total
177	v	$vw+v+1$	B7	88	2D	$[1,V],[W,W^4],[X^{16},X]$	F20AEC5DBEC480E8	0227AAC18E21CE59	0DCB274FC0980C8A	8	66	103
11D	v	$(v+1)w+v$	D6	98	C4	$[1,V],[1,W],[X,X^{16}]$	5339882404D70232	8C76181BAC0802EF	EFA440ACEA187460	8E	67	105
1CF	v	$(v+1)w+1$	3D	ED	9	$[1,V],[1,W],[X,X^{16}]$	0CC82C469FE022AD	72D6A00BE464A253	520B02BAB98646E4	81	67	106
187	v	$(v+1)w+v$	AB	74	57	$[V,V^2],[W,W^4],[X^{16},X]$	0327531B0C8D127C	7F51A9F16169F373	F56E5CA0F2968A8C	8	66	106
1E7	$v+1$	$(v+1)w+1$	84	31	F1	$[1,V],[W,W^4],[X^{16},X]$	91E4C9417D8AA092	F8A5FACDC8E734B5	8598346E041CD8F8	8	66	106
19F	$v+1$	$(v+1)w$	FA	23	73	$[V,V^2],[W,W^4],[X^{16},X]$	A68B17476596717A	43E155D129DB4D67	438496B8F84D5DE0	72	66	107
13F	v	$(v+1)w$	94	28	15	$[1,V],[1,W^4],[X^{16},X]$	0ACC409B041BE658	1420C25D7C08FCD9	011E10437454CDD9	7A	67	109
1BD	$v+1$	vw	26	53	8D	$[1,V^2],[1,W],[X^{16},X]$	5D84028CBB93CAD2	C6B45C53508620B1	45E589B90CDA65C4	89	67	109
1A9	v	$(v+1)w+v$	C7	F6	52	$[1,V],[1,W],[X,X^{16}]$	24F940D16E60791A	422024A974A4DCDB	83A31CD507280ADB	68	67	110
18B	$v+1$	$(v+1)w$	77	C1	F5	$[1,V^2],[1,W^4],[X,X^{16}]$	BF539B1BE6B12823	30BEEC13EE4C26CB	264CF2FC9B8FB78E	61	67	111
1A3	$v+1$	$(v+1)w+1$	29	BB	27	$[1,V],[1,W^4],[X^{16},X]$	8480BD26C2339DB8	40BA223556C0F2E1	E1FAC8996F3023C1	16	68	111
11B	$v+1$	vw	BD	5C	FE	$[V,V^2],[W,W^4],[X^{16},X]$	12EBED427EB22204	E77163E19B01614F	F87C3E1F8FC7E3F1	63	66	111
14D	v	vw	1D	FA	F5	$[V,V^2],[W,W^4],[X,X^{16}]$	4E41B124ACDE506D	49CF2BCD513B258F	011E10437454CDD9	A6	66	111
17B	v	$(v+1)w+v$	6C	7E	2	$[1,V],[1,W],[X^{16},X]$	6660CA6AE84A1501	245414FF6CFC3C01	4A9FFC577FD73804	97	67	111
1C3	$v+1$	vw	AD	23	5A	$[V,V^2],[W,W^4],[X^{16},X]$	5622901ED1592868	9F0391BF93EDD12B	88BB13BFE534ED48	D1	66	111
15F	v	vw	1A	84	8C	$[1,V^2],[1,W^4],[X,X^{16}]$	590888EC937B02B4	60AA86FB401C0291	16A7AC3C07626A5C	1F	67	112
12D	v	vw	BF	59	71	$[1,V],[1,W],[X,X^{16}]$	02955F7142644E4B	5488BA173C368035	6C99428038178 48D	5D	67	112
1B1	$v+1$	$vw+v$	CC	F3	98	$[1,V],[1,W^4],[X^{16},X]$	3D246A9D1B460489	D2CE42136402C8B7	71135076BF2EDBFF	99	67	112
18D	$v+1$	$(v+1)w+v$	4E	45	90	$[1,V^2],[W,W^4],[X^{16},X]$	11D00A938A8FFDA9	288FD6E7986BB867	18D74F6D8E3799E6	6C	66	113
12B	v	$vw+1$	EB	3D	3B	$[V,V^2],[W,W^4],[X^{16},X]$	CF9A81142B398486	1DED670F511F033D	3DBEB3014F9FEAC9	1B	66	113
171	v	$vw+v+1$	DA	F0	39	$[1,V],[W,W^4],[X^{16},X]$	2F8FE194C42802B5	6E5FAE47AA3902BF	B5F1DBE1FC57284F	E8	66	113
1F3	v	vw	71	AA	26	$[1,V],[W,W^4],[X,X^{16}]$	72806B022D666395	40FB2C4722C310C5	19A0A7807BEB58B8	E9	66	113
1F5	v	$(v+1)w+v$	5C	C	E3	$[1,V],[1,W^4],[X,X^{16}]$	BB759120AC4A3B8D	82B410BBDC466C19	6FDD1A839A7FB394	F0	67	113
169	v	$vw+1$	7F	13	2	$[1,V^2],[W,W^4],[X^{16},X]$	88807F8F2ADF1C01	40EB64FFC03DAC01	1D4E860BCE686CC0	D7	66	114
139	$v+1$	$vw+v+1$	D4	4B	13	$[1,V],[W,W^4],[X^{16},X]$	FDAD824678084B32	9EB3CC73041DBE0B	E8B74263439E1B02	F2	66	114
165	v	$vw+1$	89	73	FC	$[V,V^2],[W,W^4],[X^{16},X]$	4782288D12DD9C01	5B07F713D79D1B01	9A1A8A9F9BA2CD67	FC	66	115
1F9	v	$vw+1$	C0	B2	82	$[1,V^2],[1,W^4],[X^{16},X]$	BD288426C480FBC3	0428BA43FA248EA3	34026D099E44E22D	DF	68	117
1DD	v	$vw+v+1$	A1	8B	22	$[1,V],[W,W^4],[X^{16},X]$	D0D791DDA5FFC981	DC7D3A217E332EDD	7DD62EE52D1B32B9	FC	66	117
1D7	v	$(v+1)w$	35	73	5A	$[1,V],[W,W^4],[X^{16},X]$	D71E2F9E932014D4	505304D97EDB3CBD	A0758DF7DEB59BE0	EE	66	118

Table 4: Subset of the smallest area-optimized 16-bit S-box constructions for merged circuit implementations found using our methodology.

$t(v)$	1012F	1018F	10175	1015D
Σ	$v+1$	$v+1$	v	v
Π	$vw+1$	$vw+v$	vw	$(v+1)w+1$
Λ	$(vw+1)x$	$(vw+1)x+vw+v+1$	$((v+1)w+v)x$	$((v+1)w+1)x$
$\mathbb{F}!v$	8AA4	A477	F723	10C
$\mathbb{F}!w$	5628	6610	953F	1B79
$\mathbb{F}!x$	E432	45D1	C130	8E3D
$\mathbb{F}!y$	7FC0	A8D2	12A9	849F
Bases	$[V, V^2], [1, W^4]$ $[1, X^{16}], [Y^{256}, Y]$	$[1, V^2], [1, W]$ $[X, X^{16}], [Y^{256}, Y]$	$[1, V^2], [1, W]$ $[X, X^{16}], [1, Y^{256}]$	$[V, V^2], [1, W^4]$ $[1, X^{16}], [Y, Y^{256}]$
$\mathbf{T}^{-1}$	5604FC5A41E67B644A40A8BEF62D	481CC788160890C95C3826FA6AAC	2EB8C0C54644A226B89D3EAFE542	1040FB41564B5837AA5A8B8CF735
$\mathbf{T}$	A03F998C29ECECA261AA8C68D89B	A923247E3ED95F525A598C2463DC	0617A0B82ECAFE880051803463FD	0ED2C47C0C07040CE435D4064289
$\mathbf{M}$	9F5A116333100CB33A4604FD272A	66FEAC4F696AC9C4275E7DDCBA77	A6EAF2D22BC542875BE444ECAF5A	1BB5CC9E4C55E7F139D7E3584A5D
c	1A2E	8EA3	39F8	F9D8
Inverse	367	376	390	367
Total	1209	1230	1231	1238