Searching the Space of Tower Field Implementations of the $\mathbb{F}_{2^8}$ Inverter – with Applications to AES, Camellia, and SM4

Zihao Wei, Siwei Sun*, Lei Hu, Man Wei

State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China,
School of Cyber Security, University of Chinese Academy of Sciences,
Beijing 100049, China
E-mail: {weizihao, sunsiwei, hulei, weiman}@iie.ac.cn
*Corresponding author

René Peralta

Computer Security Division, NIST,
100 Bureau Drive, Stop 8930, Gaithersburg, MD 20899-8930, USA
E-mail: rene.peralta@nist.gov

Abstract: The tower field implementation of the $\mathbb{F}_{2^8}$ inverter is not only the key technique for compact implementations of the S-boxes of several internationally standardized block ciphers such as AES, Camellia, and SM4, but also the underlying structure many side-channel attack resistant AES implementations rely on. In this work, we conduct an exhaustive study of the tower field representations of the $\mathbb{F}_{2^8}$ inverter with normal bases by applying several state-of-the-art combinatorial logic minimization techniques. As a result, we achieve improved implementations of the AES, Camellia and SM4 S-boxes in terms of area footprint. Surprisingly, we are still able to improve the currently known most compact implementation of the AES S-box from CHES 2018 by 5.5 GE, beating the record again. For Camellia and SM4, the improvements are even more significant. The Verilog codes of our implementations of the AES, Camellia and SM4 S-boxes are openly available.

Keywords: Tower field; Inverter; S-box; AES; Camellia; SM4.

Biographical notes: Zihao Wei received his B.S. degree in computer science from Xidian University, Xian, China, in 2015. He is currently working toward the Ph.D. degree in the School of Cyber Security, University of Chinese Academy of Sciences, China. His research interest includes efficient and secure hardware implementation of cryptographic primitives and cryptanalysis of lightweight cipher.

Siwei Sun is an associate professor with the Institute of Information Engineering, Chinese Academy of Sciences, where his main research interest includes symmetric-key cryptography, automatic cryptanalysis and implementation, and cryptanalysis with quantum algorithms. He holds a Ph.D. in Computer Science from the University of Chinese Academy of Sciences, Beijing.

Lei Hu received his B.S. degree and M.S. degree from Peking University, Beijing, China, in 1988 and 1991, respectively, and received his Ph.D. degree from the Chinese Academy of Sciences in 1994. Since 2002 he has been a professor at the Chinese Academy of Sciences. His research interest includes cryptograph and information security.

Man Wei received her B.S. degree in Information and Computing Sciences from Nankai University, Tianjin, China, in 2015. She is currently working toward the Ph.D. degree in the School of Cyber Security, University of Chinese Academy of Sciences, China. Her research interest includes side channel attack and protection.

René Peralta is a Computer Scientist with the Cryptographic Technology Group at NIST, where he is mainly engaged with the Interoperable Randomness Beacons, Privacy Enhancing Technologies, Circuit Complexity, and Post-Quantum Cryptography projects. He holds a Ph.D. in Computer Science from the University of California, Berkeley. He has taught at various universities in the US, Chile, and Japan.

1 Introduction

For encrypting and authenticating the largest part of the workload of today's secure communication, symmetric-key primitives are regarded as the crypto workhorse (whereas public-key schemes are generally used for setting up the session keys). In many cases, the components of symmetric-key schemes are built on operations over finite fields. Since the symmetric-key cryptographic algorithms will eventually be implemented in software or hardware to play a role in the real world, it is of great importance to investigate how to implement their common operations efficiently Beierle et al. (2016); Boyar et al. (2013); Jean et al. (2017b); Stoffelen (2016); Li et al. (2019); Tan et al. (2020). For instance, due to the rapid development of lightweight IoT devices, ongoing efforts have been made to obtain more compact ASIC implementations of symmetric-key ciphers Banik et al. (2016a,b); Jean et al. (2017a). Just recently, the most compact implementation of the MixColumns and the S-box of AES were reported at FSE 2018 Kranz et al. (2017) and CHES 2018 Reyhani-Masoleh et al. (2018) respectively.

In this work, we focus on area-optimized implementations of the multiplicative inverse operation (and its affine equivalences) over $\mathbb{F}_{2^8}$. The AES S-box, which is affine equivalent to the $\mathbb{F}_{2^8}$ inverter, is the strongest 8×8 S-box known so far in terms of local security properties (*i.e.*, non-linearity, differential uniformity, algebraic degree, etc.). Several internationally standardized block ciphers, such as Camellia and SM4, apply variants of the AES S-box in their designs, which are all affine equivalent to the $\mathbb{F}_{2^8}$ inverter. Despite its desirable local cryptographic properties, to implement the AES S-box in ASIC with small footprint is not a trivial task. The naive approach that encodes the AES S-box as a look-up table in a hardware description language and produces the actual circuit relying on open-source or commercial CAD tools will certainly lead to unsatisfactory results for many resource constrained applications. Today's most compact ASIC implementations of the AES S-box are based on the tower field architecture, where the operations over $\mathbb{F}_{2^{2k}}$ are represented with operations over $\mathbb{F}_{2^k}$ recursively.

Moreover, several cost-effective threshold implementations of the AES S-box with resistance against side-channel attacks are built on top of the tower field architecture Bilgin

et al. (2014, 2015); Cnudde et al. (2016); Moradi et al. (2011). In threshold implementations, the most important design consideration includes the security level, number of fresh random bits required, and area consumption. Therefore, providing different implementations of the tower field structure without increasing the circuit footprint potentially offers more flexible area-randomness-security trade-off in threshold implementations.

Apart from these, breaking the AES S-box into several layers with the tower field architecture allows registers to be inserted into the middle of the computation such that the critical path can be reduced, and therefore the frequency of the system clock can be increased to boost the performance.

Related work. The tower field architecture was first proposed by Itoh and Tsujii for computing multiplicative inverse in finite fields of characteristic two Itoh and Tsujii (1988). At the beginning, it was applied in developing efficient implementations of public-key cryptographic algorithms involving inverse operations over $\mathbb{F}_{2^n}$ Guajardo and Paar (1997); Paar and Soria-Rodriguez (1997). Later, after the development of the Advanced Encryption Standard (AES) – a block cipher using an S-box affine equivalent to the $\mathbb{F}_{2^8}$ multiplicative inverter Daemen and Rijmen (2002), the tower field architecture found applications in compact hardware implementations of AES Mentens et al. (2005); Rudra et al. (2001); Satoh et al. (2001); Wolkerstorfer et al. (2002). After a series of improvements, Canright Canright (2005a,b) and Boyar et al. Boyar et al. (2013); Circuit Minimization Team (CMT) achieved the most compact implementations at the time, which has become the *de facto* standard for compact implementations of the AES S-box. Such tower field implementations of the AES S-box were also intensively applied in side-channel resistant implementations of AES to reduce resource consumption. Recently Reyhani-Masoleh et al. Reyhani-Masoleh et al. (2018) broke the record set by Canright and Boyar et al., presenting so far the most compact ASIC implementation of the AES S-box, which costs 182.25 GE under the STM 65nm CMOS technology.

Due to the strong local cryptographic properties of the AES S-box, several well known block ciphers employ affine equivalences of the AES S-box as their S-boxes, including Camellia and SM4. Therefore, the technique of tower field implementation naturally applies to these ciphers Abbasi and Afzal (2011); Bai et al. (2009); Martínez-Herrera et al. (2012, 2013); Satoh and Morioka (2003).

In tower field implementations, a sequence of field extensions starting from $\mathbb{F}_2$ and ending at $\mathbb{F}_{2^8}$ of the type $\mathbb{F}_{2^k} \subseteq \mathbb{F}_{(2^k)^{2^l}}$ is considered. At each level of the field extension, an irreducible polynomial over $\mathbb{F}_{2^k}$ and a corresponding basis of $\mathbb{F}_{(2^k)^{2^l}}$ over $\mathbb{F}_{2^k}$ have to be specified. The irreducible polynomials and bases induce a basis of $\mathbb{F}_{2^8}$ over $\mathbb{F}_2$. The tower field architecture is implemented over this new basis with proper basis transformations to maintain the original field representation. Therefore, the choices of the field extensions, the corresponding irreducible polynomials and the bases determine the overall cost of the implementation. A summary of the choices of existing work is given in Table 1 for AES, Camellia and SM4 respectively.

Contributions. As shown in Table 1, only a part of the design space of tower field implementation was explored by choosing irreducible polynomials of special forms in previous work. In particular, previous work preferred a class of parameter choices where the irreducible polynomials selected for the field extension $\mathbb{F}_{2^2} \subseteq \mathbb{F}_{2^4} \subseteq \mathbb{F}_{2^8}$ are of the form $z^2 + z + N$ and $y^2 + y + \nu$, and indeed the most well known implementations of Canright's and Boyar et al.'s schemes are in this class Boyar et al. (2013); Canright (2005b).

Table 1 Previous tower field implementations of the AES, Camellia and SM4 S-boxes, where $\mathcal{P}$ means a polynomial basis is used, $\mathcal{N}$ means a normal basis is used, and #Cases denotes the number of cases considered.

Cipher	Source	Tower field architecture and basis	#Cases
AES	Rudra et al. (2001)Wolkerstorfer et al. (2002)	$\mathbb{F}_2 \xrightarrow{w^4+w+1} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}]{y^2+y+C_1} \mathbb{F}_{2^8}$	1
	Satoh et al. (2001)	$\mathbb{F}_2 \xrightarrow[\mathcal{P}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{P}]{z^2+z+C_2} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}]{y^2+y+C_3} \mathbb{F}_{2^8}$	1
	Mentens et al. (2005)	$\mathbb{F}_2 \xrightarrow[\mathcal{P}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{P}]{z^2+z+C_2} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}]{y^2+y+\nu} \mathbb{F}_{2^8}$	64
	Canright (2005b)	$\mathbb{F}_2 \xrightarrow[\mathcal{P}/\mathcal{N}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{P}/\mathcal{N}]{z^2+z+N} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}/\mathcal{N}]{y^2+y+\nu} \mathbb{F}_{2^8}$	432
	Boyar et al. (2013)	$\mathbb{F}_2 \xrightarrow[\mathcal{N}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{N}]{z^2+z+C_4} \mathbb{F}_{2^4} \xrightarrow[\mathcal{N}]{y^2+y+C_5} \mathbb{F}_{2^8}$	1
	Reyhani-Masoleh et al. (2018)	$\mathbb{F}_2 \xrightarrow[\mathcal{N}]{w^4+w^3+w^2+w+1} \mathbb{F}_{2^4} \xrightarrow[\mathcal{N}]{y^2+y+\nu} \mathbb{F}_{2^8}$	32
Camellia	Satoh and Morioka (2003)	$\mathbb{F}_2 \xrightarrow{w^4+w+1} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}]{y^2+y+C_6} \mathbb{F}_{2^8}$	1
	Martínez-Herrera et al. (2012)	$\mathbb{F}_2 \xrightarrow[\mathcal{P}/\mathcal{N}]{w^4+w+1} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}/\mathcal{N}]{y^2+\tau y+\nu} \mathbb{F}_{2^8}$ $\mathbb{F}_2 \xrightarrow[\mathcal{P}/\mathcal{N}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{P}/\mathcal{N}]{z^2+Tz+N} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}/\mathcal{N}]{y^2+\tau y+\nu} \mathbb{F}_{2^8}$	13
SM4	Martínez-Herrera et al. (2013)	$\mathbb{F}_2 \xrightarrow[\mathcal{P}/\mathcal{N}]{w^4+w+1} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}/\mathcal{N}]{y^2+C_7y+C_8} \mathbb{F}_{2^8}$	4
	Bai et al. (2009)	$\mathbb{F}_2 \xrightarrow[\mathcal{P}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{P}]{z^2+z+C_9} \mathbb{F}_{2^4} \xrightarrow[\mathcal{P}]{y^2+y+C_{10}} \mathbb{F}_{2^8}$	1
	Abbasi and Afzal (2011)	$\mathbb{F}_2 \xrightarrow[\mathcal{N}]{w^2+w+1} \mathbb{F}_{2^2} \xrightarrow[\mathcal{N}]{z^2+z+N} \mathbb{F}_{2^4} \xrightarrow[\mathcal{N}]{y^2+y+\nu} \mathbb{F}_{2^8}$	16

The preference for this special class is reasonable, since with these choices of parameters, the implementations of some subcomponents of the circuit are free. Despite this heuristic, there is no concrete evidence that this configuration will result in optimal implementations. Therefore, we exhaustively examine all possible tower field representations under normal bases induced by irreducible polynomials (720 cases in total), and find several cases which were never considered previously enjoy the most compact implementations. Along the way, we do not only apply well-known logic minimization techniques from Canright and Boyar et al., but also resort to several state-of-the-art combinatorial logic minimization techniques Fuhs and Schneider-Kamp (2010); Jean et al. (2017b); Stoffelen (2016) developed in recent years. As a result, we beat the new record set by the work of Reyhani-Masoleh et al. Reyhani-Masoleh et al. (2018) for compact implementations of the AES S-box. Moreover, the implementations of the Camellia and SM4 S-boxes are improved significantly, and we refer reader to Table 2 for a summary of the results. Naturally, these results serve to achieve more compact implementations of AES, Camellia and SM4, and potentially provide more flexible security-randomness-area trade-offs for threshold implementations of these block ciphers. The Verilog codes of our implementation of the AES, Camellia and SM4 S-boxes are provided in the Appendix.

Organization. In Section 2, we give a brief introduction of the mathematical background of the tower field representations under different bases, as well as the frequently-used logic

Table 2: Synthesized Results, where the functionalities of some uncommon gates (e.g., `XOR3`, `OAI21`, `AOI21` etc.) are described in Section 2.

Cipher	Source	Gates used										Synthesis Results			
		XOR/XNOR	XOR3	NAND	AND	NAND3	NOR	NOT	OAI21	AOI21	OAI32	SMIC 130nm	SMIC 65nm	STM 65nm	Nangate 45nm
AES	Rudra et al. (2001)	111			58							336.33	336.75	294.50	299.33
	Satoh et al. (2001)	100			36							281.33	279.00	245.00	248.00
	Mentens et al. (2005)	96			36							272.00	270.00	237.00	240.00
	Canright (2005b)	80		34			6					226.67	220.00	200.00	200.00
	Boyar et al. (2013)	83			32							236.33	234.75	206.00	208.67
	Circuit Minimization Team (CMT)	81			32							231.67	230.25	202.00	204.67
		81		32								221.00	214.25	194.00	194.00
	Reyhani-Masoleh et al. (2018)	69		39		4	3	4				211.00	205.25	188.00	188.00
		69		31			3	5	7	1		N/A	N/A	N/A	186.00
		63	3	27			7	4			4	N/A	N/A	182.25	N/A
	Ours	69		33			8					**202.00**	**196.25**	179.00	**179.00**
		51	9	33			8					N/A	N/A	**176.75**	N/A
Camellia	Martínez-Herrera et al. (2012)	113			35			9				316.33	313.50	276.50	278.67
	Ours	68		33			8	1				**200.33**	**194.75**	**177.75**	**177.67**
SM4	Martínez-Herrera et al. (2013)	99			58			11				315.67	318.00	278.75	282.67
	Bai et al. (2009)	157			63							450.33	447.75	392.75	398.00
	Abbasi and Afzal (2011)	134			36							360.67	355.50	313.00	316.00
	Ours	66		32			9	1				**195.67**	**190.25**	**173.75**	**173.67**

gates for constructing digital circuits. Subsequently, we describe the details of the tower field implementation of the $\mathbb{F}_{2^8}$ inverter in Section 3. In Section 4, we apply state-of-the-art logic minimization techniques to a list of tower field representations of the AES, Camellia and SM4 S-box under all possible normal bases. As a result, we obtain so far the most compact implementations of these S-boxes. We conclude the paper in Section 5 and propose possible future work. The source codes of the optimized implementations for the S-boxes of AES, Camellia and SM4 are provided in the Appendix.

2 Preliminaries

We first give a brief introduction of the tower field representation. Then we list a set of gates together with their functionalities and areas. These gates will be used to implement the circuits constructed in this paper, and the overall area of each circuit will be computed accordingly.

Tower field representation. Let $\mathbb{F}_2 = \{0, 1\}$ be the finite field of two elements. It is well known that the field $\mathbb{F}_{2^k}$ with 2^k elements can be induced by an irreducible polynomial $q(x) \in \mathbb{F}_2[x]$ with degree k, i.e., $\mathbb{F}_{2^k} \cong \mathbb{F}_2[x]/(q(x))$. Assuming that X is a root of $q(x)$ over $\mathbb{F}_{2^k}$, then every element in $\mathbb{F}_{2^k}$ can be represented as an $\mathbb{F}_2$-linear combination $b_{k-1}X^{k-1} + \cdots + b_1X + b_0$ of $[X^{k-1}, \cdots, X, X^0]$, which is a *polynomial basis* of $\mathbb{F}_{2^k}$ over $\mathbb{F}_2$. To be concrete, we take $k = 8$, and we call $(b_7, \cdots, b_0)$ the bit-vector representation of $b_{k-1}X^{k-1} + \cdots + b_1X + b_0$ under the basis $[X^7, \cdots, X, X^0]$.

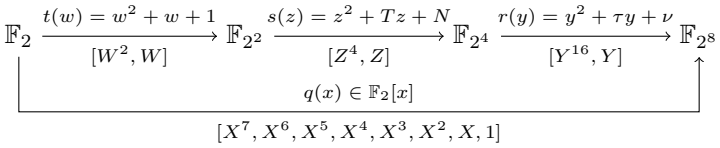

Figure 1: The tower field structure

Considering a sequence of field extensions $\mathbb{F}_2 \subseteq \mathbb{F}_{2^2} \subseteq \mathbb{F}_{2^4} \subseteq \mathbb{F}_{2^8}$ shown in Figure 1. Let $r(y) \in \mathbb{F}_{2^4}[y]$, $s(z) \in \mathbb{F}_{2^2}[z]$ and $t(w) \in \mathbb{F}_2[w]$ be irreducible polynomials over their respective fields, and let $Y \in \mathbb{F}_{2^8}$, $Z \in \mathbb{F}_{2^4}$ and $W \in \mathbb{F}_{2^2}$ be roots of $r(y)$, $s(z)$ and $t(w)$ over the corresponding fields respectively. Then we obtain a set of normal basis: $[Y^{16}, Y]$ is a basis of $\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$, $[Z^4, Z]$ is a basis of $\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$, and $[W^2, W]$ is a basis of $\mathbb{F}_{2^2}$ over $\mathbb{F}_2$. Therefore, for an element $b = b_7X^7 + \cdots + b_1X + b_0 \in \mathbb{F}_{2^8}$ we have

$$
\begin{aligned}
b &= \gamma_1 Y^{16} + \gamma_0 Y, \gamma_1, \gamma_0 \in \mathbb{F}_{2^4}, \\
\gamma_1 &= \Gamma_3 Z^4 + \Gamma_2 Z, \\
\gamma_0 &= \Gamma_1 Z^4 + \Gamma_0 Z, \Gamma_3, \Gamma_2, \Gamma_1, \Gamma_0 \in \mathbb{F}_{2^2}, \\
\Gamma_3 &= g_7 W^2 + g_6 W, \\
\Gamma_2 &= g_5 W^2 + g_4 W, \\
\Gamma_1 &= g_3 W^2 + g_2 W,
\end{aligned}
$$

$$\Gamma_0 = g_1 W^2 + g_0 W, g_i \in \mathbb{F}_2, 0 \leq i \leq 7,$$

which implies

$$\begin{aligned} b=& b_7 X^7 + \cdots + b_1 X + b_0 \\ &= g_7 W^2 Z^4 Y^{16} + g_6 W Z^4 Y^{16} + g_5 W^2 Z Y^{16} + g_4 W Z Y^{16} \\ &+ g_3 W^2 Z^4 Y + g_2 W Z^4 Y + g_1 W^2 Z Y + g_0 W Z Y. \end{aligned}$$

That is, b can be represented as $(g_7, \cdots, g_0)$ under the *tower basis*

$$\mathcal{TB} = [W^2 Z^4 Y^{16}, W Z^4 Y^{16}, W^2 Z Y^{16}, W Z Y^{16}, W^2 Z^4 Y, W Z^4 Y, W^2 Z Y, W Z Y]$$

induced by W, Z and Y. We call $(g_7, \cdots, g_0)$ the bit-vector representation of b under the tower basis. Assuming that the tower basis $\mathcal{TB}$ can be represented by the original polynomial basis with a matrix $M_t \in \mathbb{F}_2^{8\times 8}$ as

$$\mathcal{TB} = (X^7, \cdots, X^0) M_t,$$

we have

$$(b_7, \cdots, b_0)^\mathsf{T} = M_t \cdot (g_7, \cdots, g_0)^\mathsf{T} \text{ or } (g_7, \cdots, g_0)^\mathsf{T} = M_t^{-1} \cdot (b_7, \cdots, b_0)^\mathsf{T}.$$

Therefore, we can change the representations by multiplying M_t or M_t^{-1}, and we call M_t the *basis transformation matrix*.

Considering the example from AES shown in Figure 1, where $q(x)$ is the Rijndael polynomial

$$\begin{aligned} q(x) &= x^8 + x^4 + x^3 + x + 1, \\ \tau &= X^7 + X^5 + X^4 + X^3 + X^2 + 1, \\ \nu &= X^7 + X^6 + X^5, \\ T &= X^7 + X^5 + X^4 + X^3 + X^2 + 1, \\ N &= 1, \\ W &= X^7 + X^5 + X^4 + X^3 + X^2, \\ Z &= X^6 + X^4, \\ Y &= X^6 + X^3. \end{aligned}$$

Then we have $\mathcal{TB} = (X^7, \cdots, X^0) \cdot M_t$, where

$$M_t = \begin{pmatrix} 1&1&1&0&0&0&0&0 \\ 1&0&0&1&0&1&1&0 \\ 1&0&1&1&0&1&0&1 \\ 0&1&0&0&0&1&0&1 \\ 0&0&0&0&0&1&0&0 \\ 1&0&1&1&1&1&1&1 \\ 0&1&1&1&0&1&1&1 \\ 1&0&1&1&1&1&0&1 \end{pmatrix}.$$

Frequently-used gates. The circuits of this paper are eventually synthesized with the gates provided in common cell libraries. We list a set of frequently-used gates in Table 3, where

the area is measured in gate equivalence (GE), corresponding to the area of a two-input drive-strength-one NAND gate.

Note that apart from those common gates (XOR, XNOR, AND, NAND, OR, NOR, NOT) which are available in almost all CMOS technology libraries, we also list some compound gates (XOR3, NAND3, OAI21, AOI21, OAI32).

The data of STM 65nm library is collected from Reyhani-Masoleh et al.'s paper Reyhani-Masoleh et al. (2018), while the others comes from library files and databooks. The cell in blank of STM 65nm means the corresponding gate does not appear at Reyhani-Masoleh et al. (2018), and the cell labeled as N/A means the library does not support this kind of gate.

Table 3 Frequently-used gates in common CMOS technology libraries.

Gate	Area (GE)			
	SMIC 130nm	SMIC 65nm	STM 65nm	Nangate 45nm
XOR: $(a, b) \mapsto a \oplus b$	2.33	2.25	2	2
XNOR: $(a, b) \mapsto a \odot b$	2.33	2.25	2	2
XOR3: $(a, b, c) \mapsto a \oplus b \oplus c$	5.67	4.75	3.75	N/A
AND: $(a, b) \mapsto a \cdot b$	1.33	1.5	1.25	1.33
NAND: $(a, b) \mapsto \overline{a \cdot b}$	1	1	1	1
NAND3: $(a, b, c) \mapsto \overline{a \cdot b \cdot c}$	1.33	1.25	1.25	1.33
OR: $(a, b) \mapsto a \mid b$	1.33	1.5	1.25	1.33
NOR: $(a, b) \mapsto \overline{a \mid b}$	1	1	1	1
NOT: $a \mapsto \overline{a}$	0.66	0.75	0.75	0.66
OAI21: $(a, b, c) \mapsto \overline{(a \mid b) \cdot c}$	1.67	1.5		1.33
AOI21: $(a, b, c) \mapsto \overline{(a \cdot b) \mid c}$	1.67	1.5		1.33
OAI32: $(a, b, c, d, e) \mapsto \overline{(a \mid b \mid c) \cdot (d \mid e)}$	2.33	N/A	2	N/A

3 Tower Field Implementation of the $\mathbb{F}_{2^8}$ Inverter

In this section, we give an introduction to the tower field implementation of the $\mathbb{F}_{2^8}$ inverter. Please note that the derivation of these results can all be found in Canright's paper Canright (2005a,b).

Consider the field extension $\mathbb{F}_{2^4} \subseteq \mathbb{F}_{2^8}$ with an irreducible polynomial $r(y) = y^2 + \tau y + \nu \in \mathbb{F}_{2^4}[y]$. Let $Y \in \mathbb{F}_{2^8}$ be a root of $r(y)$. Then Y^{16} and Y form a normal basis, and every element $g \in \mathbb{F}_{2^8}$ can be represented as $g = \gamma_1 Y^{16} + \gamma_0 Y$, where $\gamma_1, \gamma_0 \in \mathbb{F}_{2^4}$. Let $g^{-1} = \delta_1 Y^{16} + \delta_0 Y$ with $\delta_1, \delta_0 \in \mathbb{F}_{2^4}$ be the inverse of g. By solving the equation

$$g \cdot g^{-1} = (\gamma_1 Y^{16} + \gamma_0 Y)(\delta_1 Y^{16} + \delta_0 Y) = 1$$

for δ_1 and δ_0, we obtain

$$\begin{aligned} \delta_1 &= [\gamma_1\gamma_0\tau^2 + (\gamma_1 + \gamma_0)^2\nu]^{-1}\gamma_0 \\ \delta_0 &= [\gamma_1\gamma_0\tau^2 + (\gamma_1 + \gamma_0)^2\nu]^{-1}\gamma_1. \end{aligned}$$

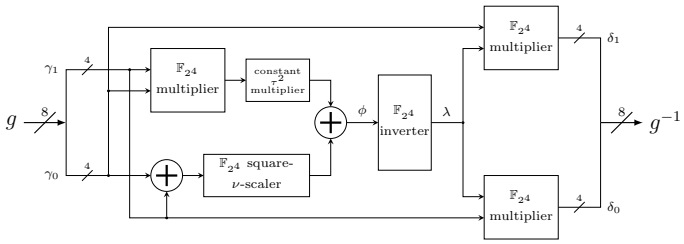

Figure 2: The $\mathbb{F}_{2^8}$ inverter

Therefore, given $r(y)$ and the basis $[Y^{16}, Y]$, we can compute $g^{-1} = (\delta_1, \delta_0)$ from $g = (\gamma_1, \gamma_0)$ using operations over $\mathbb{F}_{2^4}$, which is illustrated in Figure 2, where $\phi = \gamma_1\gamma_0\tau^2 + (\gamma_1 + \gamma_0)^2\nu$ and $\lambda = \phi^{-1}$.

Multiplication and Inverse over $\mathbb{F}_{2^4}$ Extend $\mathbb{F}_{2^2}$ to $\mathbb{F}_{2^4}$ with an irreducible polynomial $s(z) = z^2 + Tz + N \in \mathbb{F}_{2^2}[z]$, and let $Z \in \mathbb{F}_{2^4}$ be a root of $s(z)$. Then every element in $\mathbb{F}_{2^4}$ can be represented as an $\mathbb{F}_{2^2}$-linear combination of the normal basis $[Z^4, Z]$. Let $\gamma = \Gamma_1 Z^4 + \Gamma_0 Z$, and $\lambda = \Lambda_1 Z^4 + \Lambda_0 Z$, where $\Gamma_i, \Lambda_j \in \mathbb{F}_{2^2}$. Then the multiplication of γ and λ can be calculated as

$$\begin{aligned} \gamma\lambda = &\ (\Gamma_1 Z^4 + \Gamma_0 Z)(\Lambda_1 Z^4 + \Lambda_0 Z) \\ = &\ [\Gamma_1\Lambda_1 T + (\Gamma_1 + \Gamma_0)(\Lambda_1 + \Lambda_0)NT^2]Z^4 + \\ &\ [\Gamma_0\Lambda_0 T + (\Gamma_1 + \Gamma_0)(\Lambda_1 + \Lambda_0)NT^2]Z, \end{aligned} \tag{1}$$

which is illustrated in Figure 3.

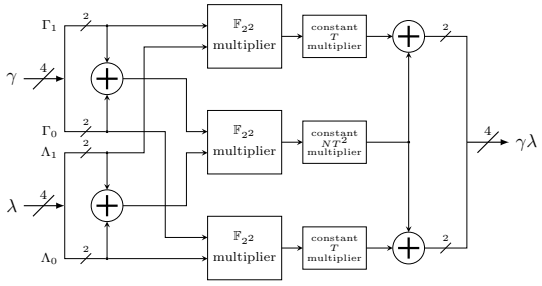

Figure 3: The $\mathbb{F}_{2^4}$ multiplier

Let $\phi = \Phi_1 Z^4 + \Phi_0 Z$ with $\Phi_i \in \mathbb{F}_{2^2}$. It can be shown that the inverse ϕ^{-1} of ϕ is

$$[\Phi_1\Phi_0 T^2 + (\Phi_1 + \Phi_0)^2 N]^{-1}\Phi_0 Z^4 + [\Phi_1\Phi_0 T^2 + (\Phi_1 + \Phi_0)^2 N]^{-1}\Phi_1 Z, \tag{2}$$

whose circuit is depicted in Figure 4.

Multiplication and Inverse over $\mathbb{F}_{2^2}$**.** Consider the field extension $\mathbb{F}_2 \subseteq \mathbb{F}_{2^2}$ with irreducible polynomial $t(w) = w^2 + w + 1 \in \mathbb{F}_2[w]$ (the only irreducible polynomial in

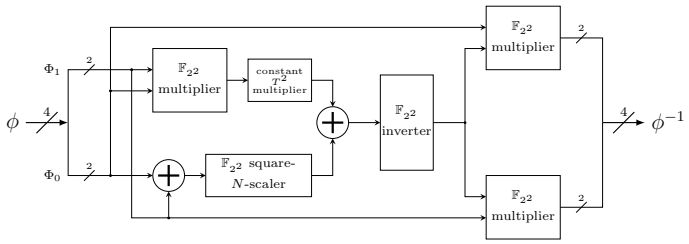

Figure 4: The $\mathbb{F}_{2^4}$ inverter

$\mathbb{F}_2[w]$). Let W be a root of $t(w)$ over $\mathbb{F}_{2^2}$. Then every element $\Gamma \in \mathbb{F}_{2^2}$ can be represented as an $\mathbb{F}_2$-linear combination $\Gamma = u_1 W^2 + u_0 W$ of the normal basis $[W^2, W]$, with $u_i \in \mathbb{F}_2$. Let $\Delta = v_1 W^2 + v_0 W$ with $v_j \in \mathbb{F}_2$ be another element in $\mathbb{F}_{2^2}$. The multiplication is given by

$$\begin{aligned} \Gamma\Delta = & \ (u_1 W^2 + u_0 W)(v_1 W^2 + v_0 W) \\ = & \ [u_1 \cdot v_1 \oplus (u_1 \oplus u_0) \cdot (v_1 \oplus v_0)] W^2 + \\ & \ [u_0 \cdot v_0 \oplus (u_1 \oplus u_0) \cdot (v_1 \oplus v_0)] W, \end{aligned} \tag{3}$$

whose implementation is shown in Figure 5. In addition, if $\Gamma\Delta = 1$, it can be shown that $v_1 = u_0$ and $v_0 = u_1$. That is, the $\mathbb{F}_{2^2}$ inverter can be implemented by swapping the two 1-bit input signals, which is free.

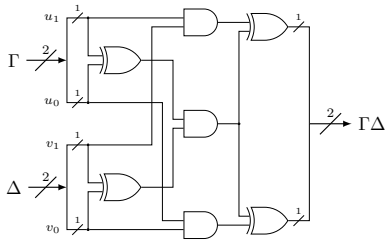

Figure 5: The $\mathbb{F}_{2^2}$ multiplier

Remark. Finally, we would like to mention two other formulas which are useful later:

$$\begin{aligned} \Gamma\Delta \cdot W &= (u_1 \cdot v_1 \oplus u_0 \cdot v_0) W^2 + [u_1 \cdot v_1 \oplus (u_1 \oplus u_0) \cdot (v_1 \oplus v_0)] W \\ \Gamma\Delta \cdot W^2 &= [u_0 \cdot v_0 \oplus (u_1 \oplus u_0) \cdot (v_1 \oplus v_0)] W^2 + (u_1 \cdot v_1 \oplus u_0 \cdot v_0) W. \end{aligned} \tag{4}$$

According to Equation 4, the implementation cost of a multiplication followed with a W (or W^2) scaler is the same as that of the multiplication $\Gamma\Delta$, which requires 4 `XOR` gates and 3 `AND` gates.

4 Applications to the S-boxes of AES, Camellia, and SM4

The S-boxes of AES, Camellia, and SM4 are all affine equivalent to the $\mathbb{F}_{2^8}$ inverter, which can be unified into the following form

$$S(b) = M_2 \cdot I^q_{\mathcal{PB}}(M_1 \cdot b \oplus C_1) \oplus C_2,\ b \in \mathbb{F}_{2^8},$$

where M_1, M_2 are 8×8 matrices over $\mathbb{F}_2$, C_1, C_2 are constant column vectors in $\mathbb{F}_2^8$, and $I^q_{\mathcal{PB}} : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ *is a function that maps the bit-vector representation of an element in* $\mathbb{F}_{2^8}$ *to the representation of its inverse in* $\mathbb{F}_{2^8}$ *under a polynomial basis of* $\mathbb{F}_{2^8}$ *over* $\mathbb{F}_2$ *induced by an irreducible polynomial* $q(x) \in \mathbb{F}_2[x]$. We refer reader to Table 4 for the concrete values of these parameters for AES, Camellia and SM4.

Table 4 The parameters of the S-boxes of AES, Camellia, and SM4, where a hexadecimal number represents an irreducible polynomial in $\mathbb{F}_2[x]$ (e.g., $x^8 + x^4 + x^3 + x + 1$ is represented by `0x11B`).

Cipher	M_1	C_1	M_2	C_2	$q(x)$
AES	$\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\\1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\\0\\0\\0\\1\\1\end{pmatrix}$	`0x11B`
Camellia*	$\begin{pmatrix}0&0&1&0&1&0&0&1\\1&0&1&0&1&0&0&0\\0&0&0&0&0&0&1&1\\0&1&1&0&0&0&0&1\\1&1&1&0&1&1&0&0\\0&1&0&1&0&1&0&1\\0&0&1&1&1&1&1&1\\0&0&0&0&0&1&1&0\end{pmatrix}$	$\begin{pmatrix}1\\1\\1\\0\\1\\1\\0\\1\end{pmatrix}$	$\begin{pmatrix}0&0&0&1&1&1&0&0\\0&1&1&1&0&0&0&0\\0&1&1&1&0&1&1&0\\1&0&0&0&1&1&0&1\\0&1&0&0&0&0&0&0\\0&0&1&0&0&1&0&1\\0&0&1&0&0&1&0&0\\0&0&0&1&0&0&0&0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\\0\\1\\1\\1\\0\end{pmatrix}$	`0x169`
SM4	$\begin{pmatrix}1&1&0&1&0&0&1&1\\1&1&1&0&1&0&0&1\\1&1&1&1&0&1&0&0\\0&1&1&1&1&0&1&0\\0&0&1&1&1&1&0&1\\1&0&0&1&1&1&1&0\\0&1&0&0&1&1&1&1\\1&0&1&0&0&1&1&1\end{pmatrix}$	$\begin{pmatrix}1\\1\\0\\1\\0\\0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1&1&0&1&0&0&1&1\\1&1&1&0&1&0&0&1\\1&1&1&1&0&1&0&0\\0&1&1&1&1&0&1&0\\0&0&1&1&1&1&0&1\\1&0&0&1&1&1&1&0\\0&1&0&0&1&1&1&1\\1&0&1&0&0&1&1&1\end{pmatrix}$	$\begin{pmatrix}1\\1\\0\\1\\0\\0\\1\\1\end{pmatrix}$	`0x1F5`

* The description of the Camellia S-box in the original specification Aoki et al. (2000) is different from ours. Reader could check the substitution table to confirm the equivalence.

However, it is difficult to implement the function $I^q_{\mathcal{PB}}$ directly with small circuit footprint. Therefore, we first implement the function $I_{\mathcal{TB}} : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ *which maps the representation of an element in* $\mathbb{F}_{2^8}$ *to the representation of its inverse element under the tower basis* $\mathcal{TB}$. According to the discussion of Section 2, we have

$$I^q_{\mathcal{PB}}(b) = M_t \cdot I_{\mathcal{TB}}(M_t^{-1} \cdot b).$$

Therefore, $S(b)$ can be implemented in practice as

$$S(b) = M_2 M_t \cdot I_{\mathcal{TB}}(M_t^{-1} M_1 \cdot b \oplus M_t^{-1} C_1) \oplus C_2,\ b \in \mathbb{F}_2^8. \tag{5}$$

Our goal is to identify a proper tower basis such that the overall circuit footprint of the implementation of $S(b)$ is minimized. Recalling the tower field architecture shown in Figure 1, the tower basis is completely determined by the three irreducible polynomials $r(y) = y^2 + \tau y + \nu \in \mathbb{F}_{2^4}[y]$, $s(z) = z^2 + Tz + N \in \mathbb{F}_{2^2}[z]$, $t(w) = w^2 + w + 1 \in \mathbb{F}_2[w]$, and their roots Y, Z and W. Therefore, the 2^{15} possible choices of τ, ν, T, N, Y, Z, and W form the overall design space[1], in which there are only 720 valid cases (we discard equivalent classes and non-irreducible polynomials).

Concretely, there are six possibilities for (T, N) making $s(z)$ irreducible, and they are $\{(1, W), (1, W^2), (W, 1), (W, W), (W^2, 1), (W^2, W^2)\}$. For each possible choice of (T, N), 120 cases of (τ, ν) can be identified such that $r(y)$ is irreducible. We can choose either one of the two roots for W, Z and Y because the other roots are exactly W^2, Z^4 and Y^{16}. So altogether there are $6 \times 120 \times 1 \times 1 \times 1 = 720$ valid cases. We exhaust all these cases for AES, Camellia and SM4 and list the optimal parameter choices in terms of compactness in Table 5.

Table 5 Optimal choices of parameters for AES, Camellia and SM4 in terms of compactness. The parameters are given with their polynomial basis representations (e.g., $X^7 + X^5 + X^4 + X^3 + X^2$ is represented by `0xBC`).

Cipher	W	T	N	Z	τ	ν	Y
AES	0xBC	0xBC	0x01	0xB0	0xBD	0x5C	0xF4
Camellia	0x7E	0x7E	0x01	0x15	0x01	0x06	0x02
SM4	0x5C	0x5C	0x01	0x7A	0x77	0x27	0x66

According to Table 5, for the parameters of AES, we have the following relationship:

$$T = W, N = 1, \tau = T \cdot Z^4 + T \cdot Z, \nu = 0 \cdot Z^4 + T^2 \cdot Z. \tag{6}$$

Similarly, for Camellia, we have $T = W, N = 1, \tau = T^2 \cdot Z^4 + T^2 \cdot Z, \nu = T \cdot Z^4 + 0 \cdot Z$, and for SM4, we have $T = W, N = 1, \tau = T^2 \cdot Z^4 + 1 \cdot Z, \nu = T \cdot Z^4 + T^2 \cdot Z$. In what follows, we focus on the optimization of the AES S-box with the optimal parameter we identified as an example. For Camellia, SM4 and other parameters, the same procedure is performed.

4.1 Optimized Implementation of the $\mathbb{F}_{2^4}$ Multiplier, τ^2 Multiplier, and the Square-ν-scaler as a Whole

Before giving the optimized implementation, we unfold the circuits of the $\mathbb{F}_{2^4}$ multiplier, τ^2 multiplier, and square-ν-scaler one by one without any optimization. Based on these unfolded circuits, we reduce their areas by applying several logic minimization techniques with necessary tweaks.

$\mathbb{F}_{2^4}$ Multiplier. By plugging Figure 5 into Figure 3, we obtain the gate-level circuit of the $\mathbb{F}_{2^4}$ multiplier shown in Figure 6, which can also be derived by substituting Equation 3 and 4 into Equation 1.

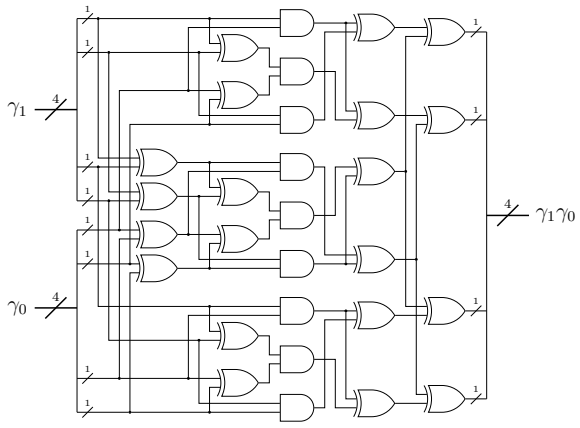

Figure 6: $\mathbb{F}_{2^4}$ multiplier

τ^2 **multiplier.** According to Table 5 and Equation 6, we have $\tau = TZ^4 + TZ = WZ^4 + WZ$ and $\tau^2 = Z^4 + Z$. Let $\alpha = (a_3W^2 + a_2W)Z^4 + (a_1W^2 + a_0W)Z$ be an arbitrary element in $\mathbb{F}_{2^4}$. We have

$$\alpha\tau^2 = [(a_3 \oplus a_2)W^2 + a_3W]Z^4 + [(a_1 \oplus a_0)W^2 + a_1W]Z,$$

leading to the gate-level circuit of the τ^2 multiplier shown in Figure 7.

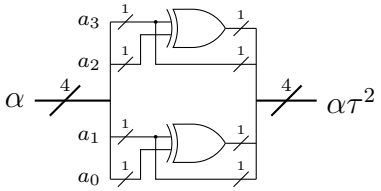

Figure 7: τ^2 multiplier

$\mathbb{F}_{2^4}$ **square-ν-scaler.** According to Table 5 and Equation 6, $\nu = W^2Z$. Let $\alpha = (a_3W^2 + a_2W)Z^4 + (a_1W^2 + a_0W)Z$ be an arbitrary element in $\mathbb{F}_{2^4}$. Then $\alpha^2\nu$ can be computed as $\alpha^2\nu = [(a_3 + a_1)W^2 + (a_3 + a_2 + a_1 + a_0)W]Z^4 + [(a_1 + a_0)W^2 + a_0W]Z$, whose gate-level circuit is shown in Figure 8.

Now, we can assemble the $\mathbb{F}_{2^4}$ multiplier, τ^2 multiplier, and square-ν-scaler to obtain a part of the $\mathbb{F}_{2^8}$ inverter according to Figure 2, which gives the circuit shown in Figure 9. According to Equation 4, this circuit can be partially optimized with some tweaks on the eight XOR gates appearing at the lower part of Figure 9, leading to the circuit shown in Figure 10. Subsequently, by applying the formula

$$a \cdot b \oplus a \oplus b = a \mid b \tag{7}$$

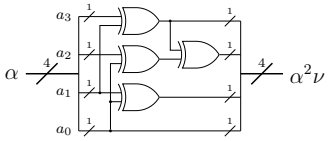

Figure 8: $\mathbb{F}_{2^4}$ square-ν-scaler

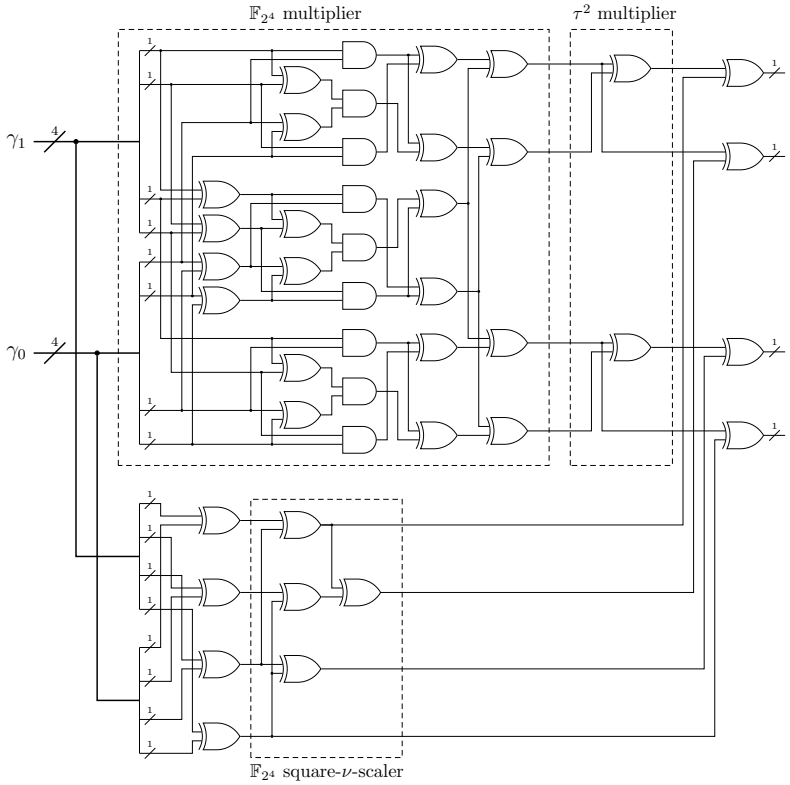

Figure 9: A part of the $\mathbb{F}_{2^8}$ inverter (unoptimized)

we can transform the circuit shown in Figure 10 into the circuit presented in Figure 11. According to Equation 7, at best, we can replace two XOR gates and one AND gate by a single OR gate. This happens for the gates marked by number 3. However, when some intermediate value of the computation $a \cdot b \oplus a \oplus b$ is required, we still need to keep some intermediate gates. For example, we can only replace two XOR gates with one OR gate and keep the AND gate intact. Similarly, for the gates marked with number 1 and number 2, we can only replace one XOR gates with one OR gate. Finally, by applying the formulas $a \cdot b \oplus c \cdot d = \overline{a \cdot b} \oplus \overline{c \cdot d}$ and $a \cdot b \oplus c \mid d = \overline{a \cdot b} \oplus \overline{c \mid d}$, the AND gates and OR gates can be substituted by NAND gates and NOR gates respectively.

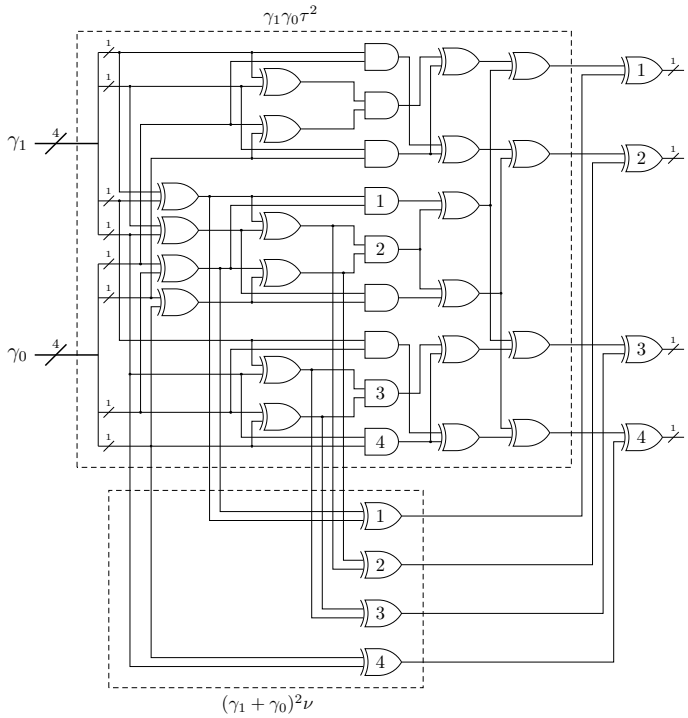

Figure 10: A part of the $\mathbb{F}_{2^8}$ inverter (partially optimized)

4.2 Optimized Implementation of the $\mathbb{F}_{2^4}$ Inverter

Based on the selected parameters for AES ($T = W$ and $N = 1$) given in Table 5 and Equation 6, Equation 2 can be simplified as

$$\phi^{-1} = [\Phi_1\Phi_0 W^2 + (\Phi_1 + \Phi_0)^2]^{-1}\Phi_0 Z^4 + [\Phi_1\Phi_0 W^2 + (\Phi_1 + \Phi_0)^2]^{-1}\Phi_1 Z. \quad (8)$$

Deviating from previous implementations Boyar et al. (2013); Canright (2005a,b), we regard the $\mathbb{F}_{2^4}$ inverter as a 4×4 S-box whose permutation table is determined by Equation 8:

$$[\texttt{0x0}, \texttt{0x8}, \texttt{0x4}, \texttt{0xC}, \texttt{0x2}, \texttt{0xF}, \texttt{0x7}, \texttt{0x6}, \texttt{0x1}, \texttt{0xD}, \texttt{0xA}, \texttt{0xE}, \texttt{0x3}, \texttt{0x9}, \texttt{0xB}, \texttt{0x5}].$$

To obtain optimized implementations of this S-box, we consider two recently proposed techniques. First, we employ Stoffelen's SAT-based technique Stoffelen (2016) to produce a circuit of the 4×4 S-box: $(x_3, x_2, x_1, x_0) \mapsto (y_3, y_2, y_1, y_0)$ with minimized gate complexity, and the result is shown below:

$$\begin{aligned}
t_1 &= \overline{x_3 \cdot x_0} & t_2 &= t_1 \mid x_2 & t_3 &= \overline{x_2 \cdot x_0} \\
t_4 &= x_1 \oplus t_3 & t_5 &= \overline{x_2 \mid t_4} & t_6 &= \overline{x_1 \cdot t_4} \\
t_7 &= x_3 \mid t_4 & t_8 &= t_7 \cdot t_2 & t_9 &= t_5 \oplus t_7
\end{aligned}$$

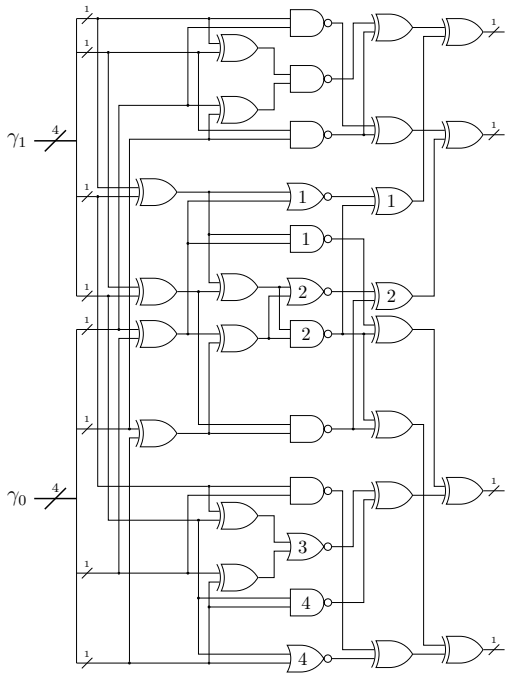

Figure 11: A part of the $\mathbb{F}_{2^8}$ inverter (optimized)

$$t_{10} = t_9 \odot x_3 \quad (y_0) \qquad t_{11} = \overline{t_6 \cdot t_8} \quad (y_2) \qquad t_{12} = \overline{t_8 \cdot x_1}$$
$$t_{13} = x_0 \odot t_{12} \quad (y_3) \qquad t_{14} = \overline{t_1 \cdot x_2} \qquad t_{15} = \overline{t_9 \cdot t_{14}} \quad (y_1),$$

which contains 4 XOR/XNOR gates, 1 AND gate, 7 NAND gates, 2 OR gates and 1 NOR gate[2]. This circuit (referred as SAT) can be further optimized manually. Since $a \cdot b = \overline{\bar{a} \mid \bar{b}}$, we can change the AND gate in t_8 to NOR gate, and negate its input signals without changing the overall functionality of the circuit. This new circuit (referred as SAT*) contains 4 XOR/XNOR gates, 7 NAND gates and 4 NOR gates (modified signals are colored in red):

$$t_1 = \overline{x_3 \cdot x_0} \qquad t_2 = \overline{t_1 \mid x_2} \qquad t_3 = \overline{x_2 \cdot x_0}$$
$$t_4 = x_1 \oplus t_3 \qquad t_5 = \overline{x_2 \mid t_4} \qquad t_6 = \overline{x_1 \cdot t_4}$$
$$t_7 = \overline{x_3 \mid t_4} \qquad t_8 = \overline{t_7 \mid t_2} \qquad t_9 = t_5 \odot t_7$$
$$t_{10} = t_9 \odot x_3 \quad (y_0) \qquad t_{11} = \overline{t_6 \cdot t_8} \quad (y_2) \qquad t_{12} = \overline{t_8 \cdot x_1}$$
$$t_{13} = x_0 \odot t_{12} \quad (y_3) \qquad t_{14} = \overline{t_1 \cdot x_2} \qquad t_{15} = \overline{t_9 \cdot t_{14}} \quad (y_1).$$

We also apply the LIGHTER Jean et al. (2017b) tool to the 4×4 S-box (the $\mathbb{F}_{2^4}$ inverter) for four different technology libraries, which leads to the same circuit (referred as LIGHTER) containing 7 XOR/XNOR gates, 4 NAND gates, 1 NAND3 gate, 1 NOR gate and 1 NOT gate shown in the following:

$$t_1 = x_2 \oplus x_3 \qquad t_2 = \overline{t_1 \cdot x_0 \cdot x_3} \qquad t_3 = x_1 \odot t_2$$
$$t_4 = \overline{t_3 \cdot t_1} \qquad t_5 = x_0 \oplus t_4 \qquad t_6 = \overline{x_3 \cdot t_5}$$

$$t_7 = t_1 \oplus t_6 \qquad t_8 = \overline{t_5 \mid t_7} \qquad t_9 = x_3 \oplus t_8 \quad (y_0)$$
$$t_{10} = \overline{t_9 \cdot t_3} \qquad t_{11} = t_5 \oplus t_{10} \quad (y_3) \qquad t_{12} = \overline{t_7} \quad (y_1)$$
$$t_{13} = \overline{t_{11} \cdot t_{12}} \qquad t_{14} = t_3 \odot t_{13} \quad (y_2).$$

A comparison of the above three circuits together with their synthesizing results is given in Table 6 and Table 7, from which we can see that SAT* is always the best, whose circuit is depicted in Figure 12.

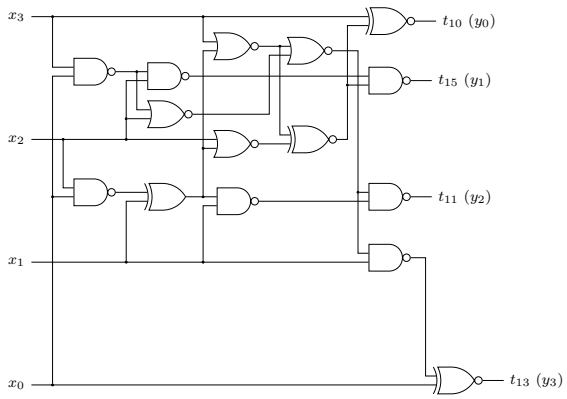

Figure 12: The optimized circuit for the $\mathbb{F}_{2^4}$ inverter (SAT*)

Table 6 Gate counts for different implementations of $\mathbb{F}_{2^4}$ inverter, where the circuit named Canright is the implementation of Equation 8 using the method in Canright (2005a,b), and the circuit named Boyar uses the method in Boyar et al. (2013).

Circuit	Gates used						
	XOR/XNOR	AND	NAND	NAND3	OR	NOR	NOT
Canright	9		8			2	
Boyar	9	6					
Boyar*	9		6				
SAT	4	1	7		2	1	
SAT*	4		7			4	
LIGHTER	7		4	1		1	1

4.3 Optimized Implementation of the Two $\mathbb{F}_{2^4}$ Multipliers with 4-bit Common Input

Observing Figure 2, the $\mathbb{F}_{2^8}$ inverter contains three $\mathbb{F}_{2^4}$ multipliers, from which we can identify three pairs of $\mathbb{F}_{2^4}$ multipliers such that each pair shares a 4-bit input signal: the leftmost $\mathbb{F}_{2^4}$ multiplier and the rightmost upper $\mathbb{F}_{2^4}$ multiplier, the leftmost $\mathbb{F}_{2^4}$ multiplier and the rightmost lower $\mathbb{F}_{2^4}$ multiplier, and the two rightmost $\mathbb{F}_{2^4}$ multipliers. It is shown

Table 7 Synthesized results for different implementations of the $\mathbb{F}_{2^4}$ inverter

Circuit	Synthesis results			
	SMIC 130nm	SMIC 65nm	STM 65nm	Nangate 45nm
Canright	30.97	30.25	28.00	28.00
Boyar	28.95	29.25	25.50	25.98
Boyar*	26.97	26.25	24.00	24.00
SAT	21.31	21.50	20.25	19.99
SAT*	**20.32**	**20.00**	**19.00**	**19.00**
LIGHTER	23.30	22.75	21.00	20.99

in Canright (2005a,b) that whenever two $\mathbb{F}_{2^4}$ multipliers share a common 4-bit input signal, some XOR gates can be saved via signal reuse.

As an example, we unfold the two rightmost $\mathbb{F}_{2^4}$ multipliers in Figure 2 according to Figure 6, and the schematic is shown in Figure 13. By observing Figure 13 carefully, we can spot some outputs of XOR gates which are computed twice in the circuit Canright (2005a,b) (labeled with same numbers in the figure). Therefore, for each pair of $\mathbb{F}_{2^4}$ multipliers sharing a 4-bit input signal, we can remove 5 XOR gates by signal reuse. Therefore, 3 pairs of $\mathbb{F}_{2^4}$ multipliers with shared input signals in total save $5 \times 3 = 15$ XOR gates.

4.4 Optimized Implementation of the Input and Output Affine Parts

According to Equation 5, before going into the $\mathbb{F}_{2^8}$ inverter $I_{\mathcal{TB}}(\cdot)$, the 8-bit input signal of the AES S-box first goes through an affine transformation

$$b \mapsto g = M_t^{-1} M_1 \cdot b \oplus M_t^{-1} C_1,$$

which then spawns 18 1-bit signals (see Figure 11) subsequently fed into some non-linear gates (NAND, NOR). The transformation from the 8 1-bit input signals to the 18 1-bit signals is affine, and can be represented as an 18×8 matrix

$$U = \begin{pmatrix} 0\,1\,0\,0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,1\,1\,0\,0\,1 \\ 0\,0\,1\,1\,1\,1\,1\,1\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0 \\ 0\,1\,0\,0\,1\,0\,0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,0 \\ 0\,1\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,1\,1\,1\,1\,0 \\ 0\,1\,1\,0\,0\,1\,0\,1\,1\,1\,1\,0\,1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{pmatrix}^{\mathsf{T}},$$

with the constant vector expanded from $M_t^{-1} C_1$

$$C = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{pmatrix}^{\mathsf{T}}.$$

By applying the SAT-based method for solving SLP (Shortest Linear Straight-Line Program) Fuhs and Schneider-Kamp (2010), we obtain the optimal implementation of U, which costs 19 XOR gates[3]:

$$y_{17} = x_0 \quad (y_{17}) \qquad t_1 = x_7 \oplus x_4 \quad (y_6) \qquad t_2 = x_3 \oplus x_1$$

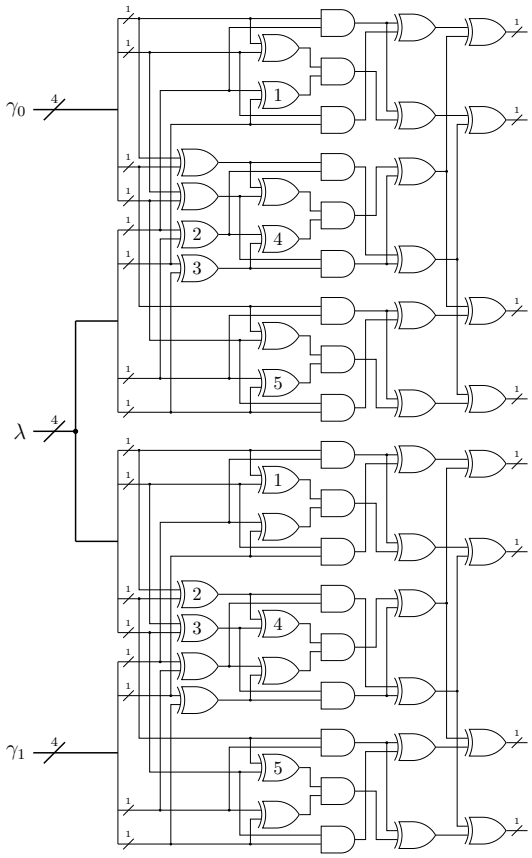

Figure 13: The two rightmost $\mathbb{F}_{2^4}$ multipliers with a shared 4-bit input

$$
\begin{array}{lll}
t_3 = x_2 \oplus t_2 & t_4 = x_6 \oplus t_3 \quad (y_7) & t_5 = x_0 \oplus t_4 \quad (y_{15}) \\
t_6 = x_5 \oplus t_3 \quad (y_1) & t_7 = t_5 \oplus t_6 \quad (y_{14}) & t_8 = x_4 \oplus t_7 \quad (y_{13}) \\
t_9 = t_1 \oplus t_2 \quad (y_9) & t_{10} = t_1 \oplus t_8 \quad (y_{11}) & t_{11} = x_0 \oplus t_9 \quad (y_{16}) \\
t_{12} = x_4 \oplus x_2 \quad (y_2) & t_{13} = x_1 \oplus t_7 \quad (y_{10}) & t_{14} = x_7 \oplus x_2 \quad (y_4) \\
t_{15} = t_{13} \oplus t_{14} \quad (y_{12}) & t_{16} = x_7 \oplus x_1 \quad (y_0) & t_{17} = t_{12} \oplus t_{16} \quad (y_8) \\
t_{18} = t_6 \oplus t_9 \quad (y_3) & t_{19} = t_7 \oplus t_{11} \quad (y_5) &
\end{array}
$$

where x_i's are the input signals, t_i's are intermediate signals, and y_i's are output signals.

Similarly, according to Equation 5, at the output end of the AES S-box, the 8-bit output of the two rightmost $\mathbb{F}_{2^4}$ multipliers (see Figure 2 and Figure 14) is transformed by the affine mapping $M_2 M_t(\cdot) \oplus C_2$ to recover the polynomial basis representation. Observing Figure 14, the 8 input bits of the affine mapping (also the output bits of the two $\mathbb{F}_{2^4}$ multipliers) are originated from the 18 output bits of the NAND gates. Moreover, only XOR gates are involved to generate the 8 input bits of the affine mapping from these 18 bits. Therefore, the mapping from the 18 output bits of the NAND gates to the 8 output bits of the

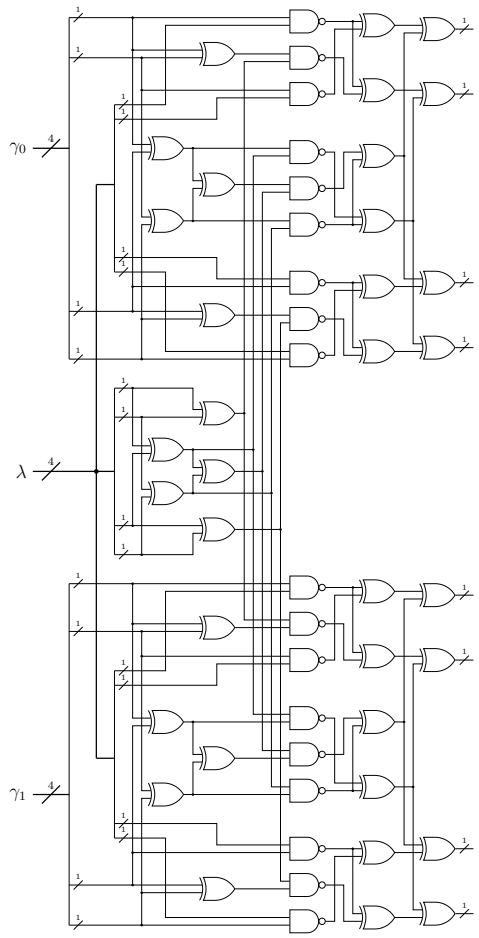

Figure 14: The circuit for bottom part

S-box is affine, which can be described by the following matrix

$$B = \begin{pmatrix} 0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1 \\ 0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0 \\ 0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,1\,1 \\ 0\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,1\,0\,1 \\ 1\,0\,1\,0\,1\,1\,1\,1\,0\,1\,1\,0\,1\,0\,1\,0\,1\,1 \\ 0\,0\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0\,1\,1\,0\,1 \\ 0\,0\,0\,0\,1\,1\,0\,1\,1\,1\,1\,1\,0\,0\,0\,1\,1\,0 \end{pmatrix}.$$

Observing Figure 14, there are two layers of XOR gates following the 18 NAND gates. According these two layers of XOR gates, we decompose the whole output affine transformation (from the 18 output bits of the 18 NAND gates to the 8 output bits of the AES S-box) into two parts. The first part maps the 18 output bits of the 18 NAND gates to the 12 output bits of the first layer of 12 XOR gates, which can be implemented with in total

12 XOR gates as shown in Figure 14. The second part maps the 12 output bits of the 12 XOR gates to the 8 output bits of the S-box, and its matrix representation B' is given in the following:

$$B' = \begin{pmatrix} 0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0 \\ 0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \\ 0\,1\,0\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\ 1\,1\,0\,1\,1\,0\,1\,0\,1\,1\,0\,1 \\ 0\,0\,0\,1\,0\,1\,0\,0\,1\,1\,1\,1 \\ 0\,0\,0\,1\,0\,1\,1\,0\,0\,0\,1\,0 \end{pmatrix},$$

Again, by applying the SAT based method for SLP Fuhs and Schneider-Kamp (2010) and taking C_2 into account, the affine transformation involving B' and constant addition at the output end can be realized as follows, requiring 17 XOR/XNOR gates[4]:

$$\begin{array}{lll} t_1 = x_2 \oplus x_0 & t_2 = x_{10} \oplus t_1 & t_3 = x_8 \oplus t_2 \quad (y_7) \\ t_4 = x_5 \oplus x_1 & t_5 = x_5 \oplus x_3 & t_6 = x_7 \oplus t_5 \\ t_7 = x_8 \oplus x_6 & t_8 = x_2 \oplus t_3 & t_9 = x_4 \oplus t_8 \quad (y_4) \\ t_{10} = t_1 \oplus t_5 & t_{11} = x_9 \odot t_6 \quad (y_5) & t_{12} = t_4 \odot t_7 \quad (y_0) \\ t_{13} = t_3 \oplus t_6 & t_{14} = x_{11} \oplus t_{13} \quad (y_2) & t_{15} = t_1 \odot t_9 \quad (y_6) \\ t_{16} = t_{10} \oplus t_{12} \quad (y_1) & t_{17} = t_4 \oplus t_9 \quad (y_3) & \end{array}$$

where x_i's are the input signals, t_i's are intermediate signals, and y_i's are output signals.

4.5 Overall Implementation Results and Comparison

We synthesis the optimized implementations of the S-boxes (AES, Camellia, SM4) using Synopsys Design Compiler 2014 (DC 2014) with four technology libraries, and the synthesized results[5] together with their technology-independent gate counts are listed in Table 2.

To make the full use of the libraries to save the circuit area, Reyhani-Masoleh et. al.'s implementations Reyhani-Masoleh et al. (2018) exploit certain compound gates in specific libraries (e.g., XOR3, NAND3, OAI21, AOI21, OAI32), which are not universally available in all technology libraries. For example, the optimal implementation offered by Reyhani-Masoleh et al. (2018) employs XOR3 and OAI32 gates, reaching 182.25 GE under the STM 65nm CMOS technology.

According to the results shown in Table 2, our implementation requires only 179 GE, which beats the record set by Reyhani-Masoleh et al. (2018) even without using any compound gates. Moreover, when the area of one XOR3 gate is smaller than the area of two XOR gates in underlying technology library, the compound gates XOR3 can be applied in our design to take the place of some standard XOR gates. With this improvements, the area of our implementation of the AES S-box can be further reduced to 176.75 GE. For the S-boxes of Camellia and SM4, it can be seen from Table 2 that the improvements are even more obvious.

5 Conclusion

By applying state-of-the-art combinatorial logic minimization techniques to an exhaustive list of tower field representations of the AES, Camellia, and SM4 S-boxes with normal bases, we identify so far the most compact implementations of these S-boxes. The results obtained in this work can be used in compact and threshold implementations of AES, Camellia, and SM4. As a potential further work, it is interesting to see how to apply similar techniques to obtain compact implementations of combined S-box/inverse S-box designs for AES, Camellia, and SM4. Moreover, this work only focus on minimizing the circuit area, it is of equal importance to investigate how to reduce the depth of the circuit as in Li et al. (2019); Reyhani-Masoleh et al. (2018).

Acknowledgement

The work is supported by the National Key R&D Program of China (Grant No. 2018YFA0704704), the Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ20180102), the National Natural Science Foundation of China (61772519, 61732021, 61802400, 61802399), and the Youth Innovation Promotion Association of Chinese Academy of Sciences.

References

Abbasi, I. and Afzal, M. (2011) 'A compact S-Box design for SMS4 block cipher', *IT Convergence and Services*, Vol. 107, pp.641–658.

Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J. and Tokita, T. (2000) 'Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis', *Selected Areas in Cryptography*, pp.39–56.

Bai, X., Xu, Y. and Guo, L. (2009) 'Securing SMS4 cipher against differential power analysis and its VLSI implementation', *IEEE Singapore International Conference on Communication Systems*, pp.167–172.

Banik, S., Bogdanov, A. and Minematsu, K. (2016a) 'Low-area hardware implementations of CLOC, SILC and AES-OTR', in *2016 IEEE International Symposium on Hardware Oriented Security and Trust*, HOST 2016, McLean, VA, USA, May 3-5, 2016, pp.71–74.

Banik, S., Bogdanov, A. and Regazzoni, F. (2016b) 'Atomic-AES: A compact implementation of the AES encryption/decryption core', in *Progress in Cryptology - INDOCRYPT 2016*, 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings, pp.173–190.

Beierle, C., Kranz, T. and Leander, G. (2016) 'Lightweight multiplication in GF(2^n) with applications to MDS matrices', in *Advances in Cryptology - CRYPTO 2016*, 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I, pp.625–653.

Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V. and Rijmen, V. (2014) 'A more efficient AES threshold implementation', in *Progress in Cryptology - AFRICACRYPT 2014*, 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings, pp.267–284.

Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V. and Rijmen, V. (2015), 'Trade-offs for threshold implementations illustrated on AES', *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 34, No. 7, pp.1188–1200.

Boyar, J., Matthews, P. and Peralta, R. (2013) 'Logic minimization techniques with applications to cryptology', *J. Cryptology*, Vol. 26, No. 2, pp.280–312.

Canright, D. (2005a) *A very compact Rijndael S-box*, Technical report NPS-MA-05-001, Naval Postgraduate School. https://www.researchgate.net/publication/235155631_A_Very_Compact_Rijndael_S-box

Canright, D. (2005b) 'A very compact S-Box for AES', in *Cryptographic Hardware and Embedded Systems - CHES 2005*, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings, pp.441–455.

Circuit Minimization Team (CMT), http://www.cs.yale.edu/homes/peralta/CircuitStuff/CMT.html

Cnudde, T. D., Reparaz, O., Bilgin, B., Nikova, S., Nikov, V. and Rijmen, V. (2016) 'Masking AES with $d + 1$ shares in hardware', in *Cryptographic Hardware and Embedded Systems - CHES 2016*, 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings, pp.194–212.

Daemen, J. and Rijmen, V. (2002) *The Design of Rijndael: AES - The Advanced Encryption Standard*, Information Security and Cryptography, Springer.

Fuhs, C. and Schneider-Kamp, P. (2010) 'Synthesizing shortest linear straight-line programs over GF(2) using SAT', in *Theory and Applications of Satisfiability Testing - SAT 2010*, 13th International Conference, SAT 2010, Edinburgh, UK, July 11-14, 2010. Proceedings, pp.71–84.

Guajardo, J. and Paar, C. (1997) 'Efficient algorithms for elliptic curve cryptosystems', in *Advances in Cryptology - CRYPTO '97*, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings, pp.342–356.

Itoh, T. and Tsujii, S. (1988) 'A fast algorithm for computing multiplicative inverses in GF(2^m)) using normal bases', *Inf. Comput.* Vol. 78, No. 3, pp.171–177.

Jean, J., Moradi, A., Peyrin, T. and Sasdrich, P. (2017a) 'Bit-sliding: A generic technique for bit-serial implementations of SPN-based primitives - applications to AES, PRESENT and SKINNY', in *Cryptographic Hardware and Embedded Systems - CHES 2017*, 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, pp.687–707.

Jean, J., Peyrin, T., Sim, S. M. and Tourteaux, J. (2017b) 'Optimizing implementations of lightweight building blocks', *IACR Trans. Symmetric Cryptol.* Vol. 2017, No. 4, pp.130–168.

Kranz, T., Leander, G., Stoffelen, K. and Wiemer, F. (2017) 'Shorter linear straight-line programs for MDS matrices', *IACR Trans. Symmetric Cryptol.* Vol. 2017, No. 4, pp.188–211.

Li, S., Sun, S., Li, C., Wei, Z. and Hu, L. (2019) 'Constructing low-latency involutory MDS matrices with lightweight circuits', *IACR Trans. Symmetric Cryptol.* Vol. 2019, No. 1, pp.84–117.

Li, S., Sun, S., Shi, D., Li, C. and Hu, L. (2019) 'Lightweight Iterative MDS Matrices: How Small Can We Go?', *IACR Trans. Symmetric Cryptol.* Vol. 2019, No. 4, pp.147–170.

Tan, Q. and Peyrin, T. (2020) 'Improved Heuristics for Short Linear Programs', *IACR Trans. Cryptogr. Hardw. Embed. Syst.* Vol. 2020, No. 1, pp.203–230.

Martínez-Herrera, A. F., Mex-Perera, J. C. and Nolazco-Flores, J. A. (2012) 'Some representations of the S-Box of Camellia in GF(((2^2)^2)^2)', in *Cryptology and Network Security*, 11th International Conference, CANS 2012, Darmstadt, Germany, December 12-14, 2012. Proceedings, pp.296–309.

Martínez-Herrera, A. F., Mex-Perera, J. C. and Nolazco-Flores, J. A. (2013) 'Merging the Camellia, SMS4 and AES S-Boxes in a single S-Box with composite bases', in *Information Security*, 16th International Conference, ISC 2013, Dallas, Texas, USA, November 13-15, 2013, Proceedings, pp.209–217.

Mentens, N., Batina, L., Preneel, B. and Verbauwhede, I. (2005) 'A systematic evaluation of compact hardware implementations for the Rijndael S-Box', in *Topics in Cryptology - CT-RSA 2005*, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings, pp.323–333.

Moradi, A., Poschmann, A., Ling, S., Paar, C. and Wang, H. (2011) 'Pushing the limits: A very compact and a threshold implementation of AES', in *Advances in Cryptology - EUROCRYPT 2011*, 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011, Proceedings, pp.69–88.

Paar, C. and Soria-Rodriguez, P. (1997) 'Fast arithmetic architectures for public-key algorithms over Galois Fields GF($(2^n)^m$)', in *Advances in Cryptology - EUROCRYPT '97*, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding, pp.363–378.

Reyhani-Masoleh, A., Taha, M. M. I. and Ashmawy, D. (2018) 'Smashing the implementation records of AES s-box', *IACR Trans. Cryptogr. Hardw. Embed. Syst.* Vol. 2018, No. 2, pp.298–336.

Rudra, A., Dubey, P. K., Jutla, C. S., Kumar, V., Rao, J. R. and Rohatgi, P. (2001) 'Efficient Rijndael encryption implementation with composite field arithmetic', in *Cryptographic Hardware and Embedded Systems - CHES 2001*, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings, number Generators, pp.171–184.

Satoh, A. and Morioka, S. (2003) 'Unified hardware architecture for 128-bit block ciphers AES and Camellia', in *Cryptographic Hardware and Embedded Systems - CHES 2003*, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings, pp.304–318.

Satoh, A., Morioka, S., Takano, K. and Munetoh, S. (2001) 'A compact Rijndael hardware architecture with S-Box optimization', in *Advances in Cryptology - ASIACRYPT 2001*, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings, pp.239–254.

Stoffelen, K. (2016) 'Optimizing S-Box implementations for several criteria using SAT solvers', in *Fast Software Encryption - FSE 2016*, 23rd International Conference, Bochum, Germany, March 20-23, 2016, Revised Selected Papers, pp.140–160.

Wolkerstorfer, J., Oswald, E. and Lamberger, M. (2002) 'An ASIC implementation of the AES S-Boxes', in *Topics in Cryptology - CT-RSA 2002*, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002, Proceedings, pp.67–78.

Notes

[1]There are 2^2 choices for T, 2^2 choices for N, 2^4 choices for τ, 2^4 choices for ν, and 2 choices for each of W, Z and Y whenever T, N, τ, ν are fixed.

[2]It costs about 38 minutes to obtain this 15-gate circuit on a PC. But we cannot confirm whether there is a 14-gate circuit, since we terminated the SAT solver after about two weeks' computation.

[3]It costs about 25 days on a PC to produce this result.

[4]It costs about 30 days on a PC to produce this circuit.

[5]We do not have access to the STM 65nm technology library. However, the authors of Reyhani-Masoleh et al. (2018) provide sufficient area information for the gates involved in this particular library, based on which we can extrapolate the results without any difficulty.

Appendix

All cases considered in this paper for AES and in Canright's paper Canright (2005a,b)

Table 8 Cases considered in this paper, where the cases colored in red mean they are also considered in Canright's paper Canright (2005a,b).

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
1	0x01	0xBC	0x5C	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xEE, 0xB2]	All normal
2				0xEC			[0xBF, 0xE3]	
3				0x01			[0xEF, 0xB3]	
4				0xB1			[0xBE, 0xE2]	
5				0x0C			[0x0F, 0x53]	
6				0xBC			[0x5E, 0x02]	
7				0x51			[0x0E, 0x52]	
8				0xE1			[0x5F, 0x03]	
9	0x01	0xBC	0xB0	0xB0	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xD2, 0x62]	All normal
10				0xEC			[0x3F, 0x8F]	
11				0x5D			[0x3E, 0x8E]	
12				0x01			[0xD3, 0x63]	
13				0x0C			[0x32, 0x82]	
14				0x50			[0xDF, 0x6F]	
15				0xE1			[0xDE, 0x6E]	
16				0xBD			[0x33, 0x83]	
17	0x01	0xBC	0xEC	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xDD, 0x31]	All normal
18				0xB0			[0x3D, 0xD1]	
19				0x5D			[0x3C, 0xD0]	
20				0xB1			[0xDC, 0x30]	
21				0x50			[0x8C, 0x60]	
22				0xBC			[0x6C, 0x80]	
23				0x51			[0x6D, 0x81]	
24				0xBD			[0x8D, 0x61]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
25	0x01	0xBC	0x5D	0xB0	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xA1, 0xFC]	All normal
26				0xEC			[0xFD, 0xA0]	
27				0x5D			[0x41, 0x1C]	
28				0x01			[0x1D, 0x40]	
29				0xBC			[0x4D, 0x10]	
30				0xE0			[0x11, 0x4C]	
31				0x51			[0xAD, 0xF0]	
32				0x0D			[0xF1, 0xAC]	
33	0x01	0xBC	0x01	0xB0	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xF3, 0xF2]	All normal
34				0xEC			[0xFF, 0xFE]	
35				0xED			[0x43, 0x42]	
36				0xB1			[0x4F, 0x4E]	
37				0x0C			[0xAF, 0xAE]	
38				0x50			[0xA3, 0xA2]	
39				0x51			[0x1F, 0x1E]	
40				0x0D			[0x13, 0x12]	
41	0x01	0xBC	0xED	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0x93, 0x7E]	All normal
42				0xB0			[0x7F, 0x92]	
43				0x01			[0x2F, 0xC2]	
44				0xED			[0xC3, 0x2E]	
45				0x50			[0x73, 0x9E]	
46				0xBC			[0x9F, 0x72]	
47				0x0D			[0xCF, 0x22]	
48				0xE1			[0x23, 0xCE]	
49	0x01	0xBC	0xB1	0x0C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xCD, 0x7C]	All normal
50				0x50			[0x2D, 0x9C]	
51				0xBC			[0x7D, 0xCC]	
52				0xE0			[0x9D, 0x2C]	
53				0x51			[0xC1, 0x70]	
54				0x0D			[0x21, 0x90]	
55				0xE1			[0x71, 0xC0]	
56				0xBD			[0x91, 0x20]	
57	0x01	0xBC	0x0C	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0x65, 0x69]	All normal
58				0xB0			[0xD5, 0xD9]	
59				0x01			[0x35, 0x39]	
60				0xED			[0x85, 0x89]	
61				0x0C			[0x34, 0x38]	
62				0xE0			[0x84, 0x88]	
63				0x51			[0x64, 0x68]	
64				0xBD			[0xD4, 0xD8]	
65	0x01	0xBC	0x50	0x5D	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xDA, 0x8A]	All normal
66				0x01			[0x36, 0x66]	
67				0xED			[0x3B, 0x6B]	
68				0xB1			[0xD7, 0x87]	
69				0x0C			[0xDB, 0x8B]	
70				0x50			[0x37, 0x67]	
71				0xBC			[0x3A, 0x6A]	
72				0xE0			[0xD6, 0x86]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
73	0x01	0xBC	0xBC	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xE7, 0x5B]	All normal
74				0xB0			[0xB6, 0x0A]	
75				0x5D			[0x57, 0xEB]	
76				0xB1			[0x06, 0xBA]	
77				0x0C			[0x07, 0xBB]	
78				0xE0			[0x56, 0xEA]	
79				0x0D			[0xB7, 0x0B]	
80				0xE1			[0xE6, 0x5A]	
81	0x01	0xBC	0xE0	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xE5, 0x05]	All normal
82				0xEC			[0xB9, 0x59]	
83				0x01			[0xE8, 0x08]	
84				0xB1			[0xB4, 0x54]	
85				0x50			[0xB5, 0x55]	
86				0xE0			[0xE9, 0x09]	
87				0x0D			[0xB8, 0x58]	
88				0xBD			[0xE4, 0x04]	
89	0x01	0xBC	0x51	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0x79, 0x28]	All normal
90				0xEC			[0x95, 0xC4]	
91				0x5D			[0x99, 0xC8]	
92				0xED			[0x75, 0x24]	
93				0x0C			[0x29, 0x78]	
94				0xBC			[0xC5, 0x94]	
95				0x0D			[0xC9, 0x98]	
96				0xBD			[0x25, 0x74]	
97	0x01	0xBC	0x0D	0xB0	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xC7, 0xCA]	All normal
98				0xEC			[0x77, 0x7A]	
99				0xED			[0x2B, 0x26]	
100				0xB1			[0x9B, 0x96]	
101				0xBC			[0xCB, 0xC6]	
102				0xE0			[0x7B, 0x76]	
103				0xE1			[0x27, 0x2A]	
104				0xBD			[0x97, 0x9A]	
105	0x01	0xBC	0xE1	0x5D	[0xBD, 0xBC]	[0x5D, 0x5C]	[0xFB, 0x1A]	All normal
106				0x01			[0xAB, 0x4A]	
107				0xED			[0xF7, 0x16]	
108				0xB1			[0xA7, 0x46]	
109				0x51			[0x47, 0xA6]	
110				0x0D			[0x17, 0xF6]	
111				0xE1			[0x4B, 0xAA]	
112				0xBD			[0x1B, 0xFA]	
113	0x01	0xBC	0xBD	0x5C	[0xBD, 0xBC]	[0x5D, 0x5C]	[0x49, 0xF4]	All normal
114				0xEC			[0xA9, 0x14]	
115				0x5D			[0x19, 0xA4]	
116				0xED			[0xF9, 0x44]	
117				0x50			[0x15, 0xA8]	
118				0xE0			[0xF5, 0x48]	
119				0x51			[0x45, 0xF8]	
120				0xE1			[0xA5, 0x18]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
121	0x01	0xBD	0xE0	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xE9, 0x09]	All normal
122				0x0D			[0xB8, 0x58]	
123				0x01			[0xE8, 0x08]	
124				0xEC			[0xB9, 0x59]	
125				0xB1			[0xB4, 0x54]	
126				0x5C			[0xE5, 0x05]	
127				0x50			[0xB5, 0x55]	
128				0xBD			[0xE4, 0x04]	
129	0x01	0xBD	0x0D	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x7B, 0x76]	All normal
130				0xED			[0x2B, 0x26]	
131				0xE1			[0x27, 0x2A]	
132				0xEC			[0x77, 0x7A]	
133				0xB1			[0x9B, 0x96]	
134				0xBC			[0xCB, 0xC6]	
135				0xB0			[0xC7, 0xCA]	
136				0xBD			[0x97, 0x9A]	
137	0x01	0xBD	0xED	0x0D	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xCF, 0x22]	All normal
138				0xED			[0xC3, 0x2E]	
139				0xE1			[0x23, 0xCE]	
140				0x01			[0x2F, 0xC2]	
141				0xBC			[0x9F, 0x72]	
142				0x5C			[0x93, 0x7E]	
143				0x50			[0x73, 0x9E]	
144				0xB0			[0x7F, 0x92]	
145	0x01	0xBD	0xE1	0x0D	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x17, 0xF6]	All normal
146				0xED			[0xF7, 0x16]	
147				0xE1			[0x4B, 0xAA]	
148				0x01			[0xAB, 0x4A]	
149				0xB1			[0xA7, 0x46]	
150				0x51			[0x47, 0xA6]	
151				0x5D			[0xFB, 0x1A]	
152				0xBD			[0x1B, 0xFA]	
153	0x01	0xBD	0x01	0x0D	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x13, 0x12]	All normal
154				0xED			[0x43, 0x42]	
155				0xEC			[0xFF, 0xFE]	
156				0x0C			[0xAF, 0xAE]	
157				0xB1			[0x4F, 0x4E]	
158				0x51			[0x1F, 0x1E]	
159				0x50			[0xA3, 0xA2]	
160				0xB0			[0xF3, 0xF2]	
161	0x01	0xBD	0xEC	0xB1	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xDC, 0x30]	All normal
162				0x51			[0x6D, 0x81]	
163				0xBC			[0x6C, 0x80]	
164				0x5C			[0xDD, 0x31]	
165				0x50			[0x8C, 0x60]	
166				0xB0			[0x3D, 0xD1]	
167				0x5D			[0x3C, 0xD0]	
168				0xBD			[0x8D, 0x61]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
169	0x01	0xBD	0x0C	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x84, 0x88]	All normal
170				0xED			[0x85, 0x89]	
171				0x01			[0x35, 0x39]	
172				0x0C			[0x34, 0x38]	
173				0x51			[0x64, 0x68]	
174				0x5C			[0x65, 0x69]	
175				0xB0			[0xD5, 0xD9]	
176				0xBD			[0xD4, 0xD8]	
177	0x01	0xBD	0xB1	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x9D, 0x2C]	All normal
178				0x0D			[0x21, 0x90]	
179				0xE1			[0x71, 0xC0]	
180				0x0C			[0xCD, 0x7C]	
181				0x51			[0xC1, 0x70]	
182				0xBC			[0x7D, 0xCC]	
183				0x50			[0x2D, 0x9C]	
184				0xBD			[0x91, 0x20]	
185	0x01	0xBD	0x51	0x0D	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xC9, 0x98]	All normal
186				0xED			[0x75, 0x24]	
187				0xEC			[0x95, 0xC4]	
188				0x0C			[0x29, 0x78]	
189				0xBC			[0xC5, 0x94]	
190				0x5C			[0x79, 0x28]	
191				0x5D			[0x99, 0xC8]	
192				0xBD			[0x25, 0x74]	
193	0x01	0xBD	0xBC	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x56, 0xEA]	All normal
194				0x0D			[0xB7, 0x0B]	
195				0xE1			[0xE6, 0x5A]	
196				0x0C			[0x07, 0xBB]	
197				0xB1			[0x06, 0xBA]	
198				0x5C			[0xE7, 0x5B]	
199				0xB0			[0xB6, 0x0A]	
200				0x5D			[0x57, 0xEB]	
201	0x01	0xBD	0x5C	0xE1	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x5F, 0x03]	All normal
202				0x01			[0xEF, 0xB3]	
203				0xEC			[0xBF, 0xE3]	
204				0x0C			[0x0F, 0x53]	
205				0xB1			[0xBE, 0xE2]	
206				0x51			[0x0E, 0x52]	
207				0xBC			[0x5E, 0x02]	
208				0x5C			[0xEE, 0xB2]	
209	0x01	0xBD	0x50	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xD6, 0x86]	All normal
210				0xED			[0x3B, 0x6B]	
211				0x01			[0x36, 0x66]	
212				0x0C			[0xDB, 0x8B]	
213				0xB1			[0xD7, 0x87]	
214				0xBC			[0x3A, 0x6A]	
215				0x50			[0x37, 0x67]	
216				0x5D			[0xDA, 0x8A]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
217	0x01	0xBD	0xB0	0xE1	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xDE, 0x6E]	All normal
218				0x01			[0xD3, 0x63]	
219				0xEC			[0x3F, 0x8F]	
220				0x0C			[0x32, 0x82]	
221				0x50			[0xDF, 0x6F]	
222				0xB0			[0xD2, 0x62]	
223				0x5D			[0x3E, 0x8E]	
224				0xBD			[0x33, 0x83]	
225	0x01	0xBD	0x5D	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0x11, 0x4C]	All normal
226				0x0D			[0xF1, 0xAC]	
227				0x01			[0x1D, 0x40]	
228				0xEC			[0xFD, 0xA0]	
229				0x51			[0xAD, 0xF0]	
230				0xBC			[0x4D, 0x10]	
231				0xB0			[0xA1, 0xFC]	
232				0x5D			[0x41, 0x1C]	
233	0x01	0xBD	0xBD	0xE0	[0xBD, 0xBC]	[0xE1, 0xE0]	[0xF5, 0x48]	All normal
234				0xED			[0xF9, 0x44]	
235				0xE1			[0xA5, 0x18]	
236				0xEC			[0xA9, 0x14]	
237				0x51			[0x45, 0xF8]	
238				0x5C			[0x49, 0xF4]	
239				0x50			[0x15, 0xA8]	
240				0x5D			[0x19, 0xA4]	
241	0xBC	0x01	0xB0	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xD2, 0x62]	All normal
242				0xEC			[0x3F, 0x8F]	
243				0x0C			[0x32, 0x82]	
244				0x50			[0xDF, 0x6F]	
245				0xE1			[0xDE, 0x6E]	
246				0xBD			[0x33, 0x83]	
247				0x5D			[0x3E, 0x8E]	
248				0x01			[0xD3, 0x63]	
249	0xBC	0x01	0xEC	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0x3D, 0xD1]	All normal
250				0x5C			[0xDD, 0x31]	
251				0xBC			[0x6C, 0x80]	
252				0x50			[0x8C, 0x60]	
253				0x51			[0x6D, 0x81]	
254				0xBD			[0x8D, 0x61]	
255				0x5D			[0x3C, 0xD0]	
256				0xB1			[0xDC, 0x30]	
257	0xBC	0x01	0x5C	0xEC	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xBF, 0xE3]	All normal
258				0x5C			[0xEE, 0xB2]	
259				0x0C			[0x0F, 0x53]	
260				0xBC			[0x5E, 0x02]	
261				0x51			[0x0E, 0x52]	
262				0xE1			[0x5F, 0x03]	
263				0xB1			[0xBE, 0xE2]	
264				0x01			[0xEF, 0xB3]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
265	0xBC	0x01	0x0C	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xD5, 0xD9]	All normal
266				0x5C			[0x65, 0x69]	
267				0x0C			[0x34, 0x38]	
268				0xE0			[0x84, 0x88]	
269				0x51			[0x64, 0x68]	
270				0xBD			[0xD4, 0xD8]	
271				0xED			[0x85, 0x89]	
272				0x01			[0x35, 0x39]	
273	0xBC	0x01	0xBC	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xB6, 0x0A]	All normal
274				0x5C			[0xE7, 0x5B]	
275				0x0C			[0x07, 0xBB]	
276				0xE0			[0x56, 0xEA]	
277				0xE1			[0xE6, 0x5A]	
278				0x0D			[0xB7, 0x0B]	
279				0x5D			[0x57, 0xEB]	
280				0xB1			[0x06, 0xBA]	
281	0xBC	0x01	0xE0	0xEC	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xB9, 0x59]	All normal
282				0x5C			[0xE5, 0x05]	
283				0xE0			[0xE9, 0x09]	
284				0x50			[0xB5, 0x55]	
285				0xBD			[0xE4, 0x04]	
286				0x0D			[0xB8, 0x58]	
287				0xB1			[0xB4, 0x54]	
288				0x01			[0xE8, 0x08]	
289	0xBC	0x01	0x50	0x0C	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xDB, 0x8B]	All normal
290				0xBC			[0x3A, 0x6A]	
291				0xE0			[0xD6, 0x86]	
292				0x50			[0x37, 0x67]	
293				0x5D			[0xDA, 0x8A]	
294				0xED			[0x3B, 0x6B]	
295				0xB1			[0xD7, 0x87]	
296				0x01			[0x36, 0x66]	
297	0xBC	0x01	0x51	0xEC	[0xBD, 0xBC]	[0x0C, 0xB0]	[0x95, 0xC4]	All normal
298				0x5C			[0x79, 0x28]	
299				0x0C			[0x29, 0x78]	
300				0xBC			[0xC5, 0x94]	
301				0xBD			[0x25, 0x74]	
302				0x0D			[0xC9, 0x98]	
303				0x5D			[0x99, 0xC8]	
304				0xED			[0x75, 0x24]	
305	0xBC	0x01	0xE1	0x51	[0xBD, 0xBC]	[0x0C, 0xB0]	[0x47, 0xA6]	All normal
306				0xE1			[0x4B, 0xAA]	
307				0xBD			[0x1B, 0xFA]	
308				0x0D			[0x17, 0xF6]	
309				0x5D			[0xFB, 0x1A]	
310				0xED			[0xF7, 0x16]	
311				0xB1			[0xA7, 0x46]	
312				0x01			[0xAB, 0x4A]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
313	0xBC	0x01	0xBD	0xEC	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xA9, 0x14]	All normal
314				0x5C			[0x49, 0xF4]	
315				0xE0			[0xF5, 0x48]	
316				0x50			[0x15, 0xA8]	
317				0x51			[0x45, 0xF8]	
318				0xE1			[0xA5, 0x18]	
319				0x5D			[0x19, 0xA4]	
320				0xED			[0xF9, 0x44]	
321	0xBC	0x01	0x0D	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xC7, 0xCA]	All normal
322				0xEC			[0x77, 0x7A]	
323				0xBC			[0xCB, 0xC6]	
324				0xE0			[0x7B, 0x76]	
325				0xE1			[0x27, 0x2A]	
326				0xBD			[0x97, 0x9A]	
327				0xED			[0x2B, 0x26]	
328				0xB1			[0x9B, 0x96]	
329	0xBC	0x01	0x5D	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xA1, 0xFC]	All normal
330				0xEC			[0xFD, 0xA0]	
331				0xBC			[0x4D, 0x10]	
332				0xE0			[0x11, 0x4C]	
333				0x51			[0xAD, 0xF0]	
334				0x0D			[0xF1, 0xAC]	
335				0x5D			[0x41, 0x1C]	
336				0x01			[0x1D, 0x40]	
337	0xBC	0x01	0xED	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0x7F, 0x92]	All normal
338				0x5C			[0x93, 0x7E]	
339				0xBC			[0x9F, 0x72]	
340				0x50			[0x73, 0x9E]	
341				0xE1			[0x23, 0xCE]	
342				0x0D			[0xCF, 0x22]	
343				0xED			[0xC3, 0x2E]	
344				0x01			[0x2F, 0xC2]	
345	0xBC	0x01	0xB1	0x0C	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xCD, 0x7C]	All normal
346				0xBC			[0x7D, 0xCC]	
347				0xE0			[0x9D, 0x2C]	
348				0x50			[0x2D, 0x9C]	
349				0x51			[0xC1, 0x70]	
350				0xE1			[0x71, 0xC0]	
351				0xBD			[0x91, 0x20]	
352				0x0D			[0x21, 0x90]	
353	0xBC	0x01	0x01	0xB0	[0xBD, 0xBC]	[0x0C, 0xB0]	[0xF3, 0xF2]	All normal
354				0xEC			[0xFF, 0xFE]	
355				0x0C			[0xAF, 0xAE]	
356				0x50			[0xA3, 0xA2]	
357				0x51			[0x1F, 0x1E]	
358				0x0D			[0x13, 0x12]	
359				0xED			[0x43, 0x42]	
360				0xB1			[0x4F, 0x4E]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
361	0xBD	0x01	0x50	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0x37, 0x67]	All normal
362				0xB1			[0xD7, 0x87]	
363				0xED			[0x3B, 0x6B]	
364				0x0C			[0xDB, 0x8B]	
365				0xE0			[0xD6, 0x86]	
366				0x01			[0x36, 0x66]	
367				0x5D			[0xDA, 0x8A]	
368				0xBC			[0x3A, 0x6A]	
369	0xBD	0x01	0xE1	0xE1	[0xBD, 0xBC]	[0xED, 0x50]	[0x4B, 0xAA]	All normal
370				0xB1			[0xA7, 0x46]	
371				0xED			[0xF7, 0x16]	
372				0xBD			[0x1B, 0xFA]	
373				0x01			[0xAB, 0x4A]	
374				0x51			[0x47, 0xA6]	
375				0x0D			[0x17, 0xF6]	
376				0x5D			[0xFB, 0x1A]	
377	0xBD	0x01	0xB1	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0x2D, 0x9C]	All normal
378				0xE1			[0x71, 0xC0]	
379				0xBD			[0x91, 0x20]	
380				0x0C			[0xCD, 0x7C]	
381				0xE0			[0x9D, 0x2C]	
382				0x51			[0xC1, 0x70]	
383				0x0D			[0x21, 0x90]	
384				0xBC			[0x7D, 0xCC]	
385	0xBD	0x01	0xED	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0x73, 0x9E]	All normal
386				0xE1			[0x23, 0xCE]	
387				0xED			[0xC3, 0x2E]	
388				0x5C			[0x93, 0x7E]	
389				0xB0			[0x7F, 0x92]	
390				0x01			[0x2F, 0xC2]	
391				0x0D			[0xCF, 0x22]	
392				0xBC			[0x9F, 0x72]	
393	0xBD	0x01	0xBD	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0x15, 0xA8]	All normal
394				0xE1			[0xA5, 0x18]	
395				0xED			[0xF9, 0x44]	
396				0x5C			[0x49, 0xF4]	
397				0xE0			[0xF5, 0x48]	
398				0x51			[0x45, 0xF8]	
399				0x5D			[0x19, 0xA4]	
400				0xEC			[0xA9, 0x14]	
401	0xBD	0x01	0x0C	0xED	[0xBD, 0xBC]	[0xED, 0x50]	[0x85, 0x89]	All normal
402				0xBD			[0xD4, 0xD8]	
403				0x0C			[0x34, 0x38]	
404				0x5C			[0x65, 0x69]	
405				0xE0			[0x84, 0x88]	
406				0xB0			[0xD5, 0xD9]	
407				0x01			[0x35, 0x39]	
408				0x51			[0x64, 0x68]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
409	0xBD	0x01	0x5C	0xE1	[0xBD, 0xBC]	[0xED, 0x50]	[0x5F, 0x03]	All normal
410				0xB1			[0xBE, 0xE2]	
411				0x0C			[0x0F, 0x53]	
412				0x5C			[0xEE, 0xB2]	
413				0x01			[0xEF, 0xB3]	
414				0x51			[0x0E, 0x52]	
415				0xEC			[0xBF, 0xE3]	
416				0xBC			[0x5E, 0x02]	
417	0xBD	0x01	0xE0	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0xB5, 0x55]	All normal
418				0xB1			[0xB4, 0x54]	
419				0xBD			[0xE4, 0x04]	
420				0x5C			[0xE5, 0x05]	
421				0xE0			[0xE9, 0x09]	
422				0x01			[0xE8, 0x08]	
423				0x0D			[0xB8, 0x58]	
424				0xEC			[0xB9, 0x59]	
425	0xBD	0x01	0xB0	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0xDF, 0x6F]	All normal
426				0xE1			[0xDE, 0x6E]	
427				0xBD			[0x33, 0x83]	
428				0x0C			[0x32, 0x82]	
429				0xB0			[0xD2, 0x62]	
430				0x01			[0xD3, 0x63]	
431				0x5D			[0x3E, 0x8E]	
432				0xEC			[0x3F, 0x8F]	
433	0xBD	0x01	0x01	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0xA3, 0xA2]	All normal
434				0xB1			[0x4F, 0x4E]	
435				0xED			[0x43, 0x42]	
436				0x0C			[0xAF, 0xAE]	
437				0xB0			[0xF3, 0xF2]	
438				0x51			[0x1F, 0x1E]	
439				0x0D			[0x13, 0x12]	
440				0xEC			[0xFF, 0xFE]	
441	0xBD	0x01	0x51	0xED	[0xBD, 0xBC]	[0xED, 0x50]	[0x75, 0x24]	All normal
442				0xBD			[0x25, 0x74]	
443				0x0C			[0x29, 0x78]	
444				0x5C			[0x79, 0x28]	
445				0x0D			[0xC9, 0x98]	
446				0x5D			[0x99, 0xC8]	
447				0xEC			[0x95, 0xC4]	
448				0xBC			[0xC5, 0x94]	
449	0xBD	0x01	0x0D	0xE1	[0xBD, 0xBC]	[0xED, 0x50]	[0x27, 0x2A]	All normal
450				0xB1			[0x9B, 0x96]	
451				0xED			[0x2B, 0x26]	
452				0xBD			[0x97, 0x9A]	
453				0xE0			[0x7B, 0x76]	
454				0xB0			[0xC7, 0xCA]	
455				0xEC			[0x77, 0x7A]	
456				0xBC			[0xCB, 0xC6]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
457	0xBD	0x01	0x5D	0xE0	[0xBD, 0xBC]	[0xED, 0x50]	[0x11, 0x4C]	All normal
458				0xB0			[0xA1, 0xFC]	
459				0x01			[0x1D, 0x40]	
460				0x51			[0xAD, 0xF0]	
461				0x0D			[0xF1, 0xAC]	
462				0x5D			[0x41, 0x1C]	
463				0xEC			[0xFD, 0xA0]	
464				0xBC			[0x4D, 0x10]	
465	0xBD	0x01	0xEC	0x50	[0xBD, 0xBC]	[0xED, 0x50]	[0x8C, 0x60]	All normal
466				0xB1			[0xDC, 0x30]	
467				0xBD			[0x8D, 0x61]	
468				0x5C			[0xDD, 0x31]	
469				0xB0			[0x3D, 0xD1]	
470				0x51			[0x6D, 0x81]	
471				0x5D			[0x3C, 0xD0]	
472				0xBC			[0x6C, 0x80]	
473	0xBD	0x01	0xBC	0xE1	[0xBD, 0xBC]	[0xED, 0x50]	[0xE6, 0x5A]	All normal
474				0xB1			[0x06, 0xBA]	
475				0x0C			[0x07, 0xBB]	
476				0x5C			[0xE7, 0x5B]	
477				0xE0			[0x56, 0xEA]	
478				0xB0			[0xB6, 0x0A]	
479				0x0D			[0xB7, 0x0B]	
480				0x5D			[0x57, 0xEB]	
481	0xBC	0xBC	0xB1	0x50	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x2D, 0x9C]	All normal
482				0xE1			[0x71, 0xC0]	
483				0x0D			[0x21, 0x90]	
484				0xBC			[0x7D, 0xCC]	
485				0xBD			[0x91, 0x20]	
486				0x0C			[0xCD, 0x7C]	
487				0xE0			[0x9D, 0x2C]	
488				0x51			[0xC1, 0x70]	
489	0xBC	0xBC	0x50	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xD7, 0x87]	All normal
490				0x50			[0x37, 0x67]	
491				0xBC			[0x3A, 0x6A]	
492				0x5D			[0xDA, 0x8A]	
493				0xED			[0x3B, 0x6B]	
494				0x0C			[0xDB, 0x8B]	
495				0xE0			[0xD6, 0x86]	
496				0x01			[0x36, 0x66]	
497	0xBC	0xBC	0xE1	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xA7, 0x46]	All normal
498				0xE1			[0x4B, 0xAA]	
499				0x0D			[0x17, 0xF6]	
500				0x5D			[0xFB, 0x1A]	
501				0xED			[0xF7, 0x16]	
502				0xBD			[0x1B, 0xFA]	
503				0x51			[0x47, 0xA6]	
504				0x01			[0xAB, 0x4A]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
505	0xBC	0xBC	0x0D	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x9B, 0x96]	All normal
506				0xE1			[0x27, 0x2A]	
507				0xBC			[0xCB, 0xC6]	
508				0xEC			[0x77, 0x7A]	
509				0xED			[0x2B, 0x26]	
510				0xBD			[0x97, 0x9A]	
511				0xE0			[0x7B, 0x76]	
512				0xB0			[0xC7, 0xCA]	
513	0xBC	0xBC	0xBC	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x06, 0xBA]	All normal
514				0xE1			[0xE6, 0x5A]	
515				0x0D			[0xB7, 0x0B]	
516				0x5D			[0x57, 0xEB]	
517				0x5C			[0xE7, 0x5B]	
518				0x0C			[0x07, 0xBB]	
519				0xE0			[0x56, 0xEA]	
520				0xB0			[0xB6, 0x0A]	
521	0xBC	0xBC	0x5D	0x0D	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xF1, 0xAC]	All normal
522				0xBC			[0x4D, 0x10]	
523				0x5D			[0x41, 0x1C]	
524				0xEC			[0xFD, 0xA0]	
525				0xE0			[0x11, 0x4C]	
526				0x51			[0xAD, 0xF0]	
527				0xB0			[0xA1, 0xFC]	
528				0x01			[0x1D, 0x40]	
529	0xBC	0xBC	0xEC	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xDC, 0x30]	All normal
530				0x50			[0x8C, 0x60]	
531				0xBC			[0x6C, 0x80]	
532				0x5D			[0x3C, 0xD0]	
533				0x5C			[0xDD, 0x31]	
534				0xBD			[0x8D, 0x61]	
535				0x51			[0x6D, 0x81]	
536				0xB0			[0x3D, 0xD1]	
537	0xBC	0xBC	0xED	0x50	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x73, 0x9E]	All normal
538				0xE1			[0x23, 0xCE]	
539				0x0D			[0xCF, 0x22]	
540				0xBC			[0x9F, 0x72]	
541				0xED			[0xC3, 0x2E]	
542				0x5C			[0x93, 0x7E]	
543				0xB0			[0x7F, 0x92]	
544				0x01			[0x2F, 0xC2]	
545	0xBC	0xBC	0x5C	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xBE, 0xE2]	All normal
546				0xE1			[0x5F, 0x03]	
547				0xBC			[0x5E, 0x02]	
548				0xEC			[0xBF, 0xE3]	
549				0x5C			[0xEE, 0xB2]	
550				0x0C			[0x0F, 0x53]	
551				0x51			[0x0E, 0x52]	
552				0x01			[0xEF, 0xB3]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
553	0xBC	0xBC	0xBD	0x50	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x15, 0xA8]	All normal
554				0xE1			[0xA5, 0x18]	
555				0x5D			[0x19, 0xA4]	
556				0xEC			[0xA9, 0x14]	
557				0xED			[0xF9, 0x44]	
558				0x5C			[0x49, 0xF4]	
559				0xE0			[0xF5, 0x48]	
560				0x51			[0x45, 0xF8]	
561	0xBC	0xBC	0x0C	0xED	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x85, 0x89]	All normal
562				0x5C			[0x65, 0x69]	
563				0xBD			[0xD4, 0xD8]	
564				0x0C			[0x34, 0x38]	
565				0xE0			[0x84, 0x88]	
566				0x51			[0x64, 0x68]	
567				0xB0			[0xD5, 0xD9]	
568				0x01			[0x35, 0x39]	
569	0xBC	0xBC	0xE0	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xB4, 0x54]	All normal
570				0x50			[0xB5, 0x55]	
571				0x0D			[0xB8, 0x58]	
572				0xEC			[0xB9, 0x59]	
573				0x5C			[0xE5, 0x05]	
574				0xBD			[0xE4, 0x04]	
575				0xE0			[0xE9, 0x09]	
576				0x01			[0xE8, 0x08]	
577	0xBC	0xBC	0x51	0x0D	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xC9, 0x98]	All normal
578				0xBC			[0xC5, 0x94]	
579				0x5D			[0x99, 0xC8]	
580				0xEC			[0x95, 0xC4]	
581				0xED			[0x75, 0x24]	
582				0x5C			[0x79, 0x28]	
583				0xBD			[0x25, 0x74]	
584				0x0C			[0x29, 0x78]	
585	0xBC	0xBC	0xB0	0x50	[0xBD, 0xBC]	[0x0D, 0xB1]	[0xDF, 0x6F]	All normal
586				0xE1			[0xDE, 0x6E]	
587				0x5D			[0x3E, 0x8E]	
588				0xEC			[0x3F, 0x8F]	
589				0xBD			[0x33, 0x83]	
590				0x0C			[0x32, 0x82]	
591				0xB0			[0xD2, 0x62]	
592				0x01			[0xD3, 0x63]	
593	0xBC	0xBC	0x01	0xB1	[0xBD, 0xBC]	[0x0D, 0xB1]	[0x4F, 0x4E]	All normal
594				0x50			[0xA3, 0xA2]	
595				0x0D			[0x13, 0x12]	
596				0xEC			[0xFF, 0xFE]	
597				0xED			[0x43, 0x42]	
598				0x0C			[0xAF, 0xAE]	
599				0x51			[0x1F, 0x1E]	
600				0xB0			[0xF3, 0xF2]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
601	0xBD	0xBD	0xEC	0x5C	[0xBD, 0xBC]	[0x51, 0xEC]	[0xDD, 0x31]	All normal
602				0xB0			[0x3D, 0xD1]	
603				0x51			[0x6D, 0x81]	
604				0xBD			[0x8D, 0x61]	
605				0x5D			[0x3C, 0xD0]	
606				0xB1			[0xDC, 0x30]	
607				0x50			[0x8C, 0x60]	
608				0xBC			[0x6C, 0x80]	
609	0xBD	0xBD	0x5C	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0xBF, 0xE3]	All normal
610				0x5C			[0xEE, 0xB2]	
611				0x51			[0x0E, 0x52]	
612				0xE1			[0x5F, 0x03]	
613				0xB1			[0xBE, 0xE2]	
614				0x01			[0xEF, 0xB3]	
615				0x0C			[0x0F, 0x53]	
616				0xBC			[0x5E, 0x02]	
617	0xBD	0xBD	0xB0	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0x3F, 0x8F]	All normal
618				0xB0			[0xD2, 0x62]	
619				0xBD			[0x33, 0x83]	
620				0xE1			[0xDE, 0x6E]	
621				0x5D			[0x3E, 0x8E]	
622				0x01			[0xD3, 0x63]	
623				0x0C			[0x32, 0x82]	
624				0x50			[0xDF, 0x6F]	
625	0xBD	0xBD	0x51	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0x95, 0xC4]	All normal
626				0x5C			[0x79, 0x28]	
627				0xBD			[0x25, 0x74]	
628				0x0D			[0xC9, 0x98]	
629				0x5D			[0x99, 0xC8]	
630				0xED			[0x75, 0x24]	
631				0x0C			[0x29, 0x78]	
632				0xBC			[0xC5, 0x94]	
633	0xBD	0xBD	0xBD	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0xA9, 0x14]	All normal
634				0x5C			[0x49, 0xF4]	
635				0x51			[0x45, 0xF8]	
636				0xE1			[0xA5, 0x18]	
637				0x5D			[0x19, 0xA4]	
638				0xED			[0xF9, 0x44]	
639				0xE0			[0xF5, 0x48]	
640				0x50			[0x15, 0xA8]	
641	0xBD	0xBD	0x0D	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0x77, 0x7A]	All normal
642				0xB0			[0xC7, 0xCA]	
643				0xBD			[0x97, 0x9A]	
644				0xE1			[0x27, 0x2A]	
645				0xB1			[0x9B, 0x96]	
646				0xED			[0x2B, 0x26]	
647				0xE0			[0x7B, 0x76]	
648				0xBC			[0xCB, 0xC6]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
649	0xBD	0xBD	0xE1	0x51	[0xBD, 0xBC]	[0x51, 0xEC]	[0x47, 0xA6]	All normal
650				0xBD			[0x1B, 0xFA]	
651				0x0D			[0x17, 0xF6]	
652				0xE1			[0x4B, 0xAA]	
653				0x5D			[0xFB, 0x1A]	
654				0xB1			[0xA7, 0x46]	
655				0x01			[0xAB, 0x4A]	
656				0xED			[0xF7, 0x16]	
657	0xBD	0xBD	0x5D	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0xFD, 0xA0]	All normal
658				0xB0			[0xA1, 0xFC]	
659				0x51			[0xAD, 0xF0]	
660				0x0D			[0xF1, 0xAC]	
661				0x5D			[0x41, 0x1C]	
662				0x01			[0x1D, 0x40]	
663				0xE0			[0x11, 0x4C]	
664				0xBC			[0x4D, 0x10]	
665	0xBD	0xBD	0xB1	0x51	[0xBD, 0xBC]	[0x51, 0xEC]	[0xC1, 0x70]	All normal
666				0xBD			[0x91, 0x20]	
667				0x0D			[0x21, 0x90]	
668				0xE1			[0x71, 0xC0]	
669				0x0C			[0xCD, 0x7C]	
670				0xE0			[0x9D, 0x2C]	
671				0x50			[0x2D, 0x9C]	
672				0xBC			[0x7D, 0xCC]	
673	0xBD	0xBD	0x01	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0xFF, 0xFE]	All normal
674				0xB0			[0xF3, 0xF2]	
675				0x51			[0x1F, 0x1E]	
676				0x0D			[0x13, 0x12]	
677				0xB1			[0x4F, 0x4E]	
678				0xED			[0x43, 0x42]	
679				0x0C			[0xAF, 0xAE]	
680				0x50			[0xA3, 0xA2]	
681	0xBD	0xBD	0xED	0x5C	[0xBD, 0xBC]	[0x51, 0xEC]	[0x93, 0x7E]	All normal
682				0xB0			[0x7F, 0x92]	
683				0x0D			[0xCF, 0x22]	
684				0xE1			[0x23, 0xCE]	
685				0x01			[0x2F, 0xC2]	
686				0xED			[0xC3, 0x2E]	
687				0x50			[0x73, 0x9E]	
688				0xBC			[0x9F, 0x72]	
689	0xBD	0xBD	0x0C	0x5C	[0xBD, 0xBC]	[0x51, 0xEC]	[0x65, 0x69]	All normal
690				0xB0			[0xD5, 0xD9]	
691				0x51			[0x64, 0x68]	
692				0xBD			[0xD4, 0xD8]	
693				0x01			[0x35, 0x39]	
694				0xED			[0x85, 0x89]	
695				0x0C			[0x34, 0x38]	
696				0xE0			[0x84, 0x88]	

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
697	0xBD	0xBD	0xE0	0xEC	[0xBD, 0xBC]	[0x51, 0xEC]	[0xB9, 0x59]	All normal
698				0x5C			[0xE5, 0x05]	
699				0xBD			[0xE4, 0x04]	
700				0x0D			[0xB8, 0x58]	
701				0xB1			[0xB4, 0x54]	
702				0x01			[0xE8, 0x08]	
703				0xE0			[0xE9, 0x09]	
704				0x50			[0xB5, 0x55]	
705	0xBD	0xBD	0x50	0x5D	[0xBD, 0xBC]	[0x51, 0xEC]	[0xDA, 0x8A]	All normal
706				0xB1			[0xD7, 0x87]	
707				0x01			[0x36, 0x66]	
708				0xED			[0x3B, 0x6B]	
709				0x0C			[0xDB, 0x8B]	
710				0xE0			[0xD6, 0x86]	
711				0x50			[0x37, 0x67]	
712				0xBC			[0x3A, 0x6A]	
713	0xBD	0xBD	0xBC	0x5C	[0xBD, 0xBC]	[0x51, 0xEC]	[0xE7, 0x5B]	All normal
714				0xB0			[0xB6, 0x0A]	
715				0x0D			[0xB7, 0x0B]	
716				0xE1			[0xE6, 0x5A]	
717				0x5D			[0x57, 0xEB]	
718				0xB1			[0x06, 0xBA]	
719				0x0C			[0x07, 0xBB]	
720				0xE0			[0x56, 0xEA]	

Table 9 Cases considered in Canright's paper Canright (2005a,b), where the cases colored in red mean they are also considered in this paper.

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
1					[0xBC, 0x01]	[0x5C, 0x01]	[0xF2, 0x01]	All polynomial
2					[0xBC, 0x01]	[0x5C, 0x01]	[0xF3, 0x01]	Mixed
3					[0xBC, 0x01]	[0x5C, 0x01]	[0xF3, 0xF2]	Mixed
4					[0xBC, 0x01]	[0x5D, 0x01]	[0xF2, 0x01]	Mixed
5					[0xBC, 0x01]	[0x5D, 0x01]	[0xF3, 0x01]	Mixed
6					[0xBC, 0x01]	[0x5D, 0x01]	[0xF3, 0xF2]	Mixed
7					[0xBC, 0x01]	[0x5D, 0x5C]	[0xF2, 0x01]	Mixed
8					[0xBC, 0x01]	[0x5D, 0x5C]	[0xF3, 0x01]	Mixed
9					[0xBC, 0x01]	[0x5D, 0x5C]	[0xF3, 0xF2]	Mixed
10					[0xBD, 0x01]	[0x5C, 0x01]	[0xF2, 0x01]	Mixed
11					[0xBD, 0x01]	[0x5C, 0x01]	[0xF3, 0x01]	Mixed
12					[0xBD, 0x01]	[0x5C, 0x01]	[0xF3, 0xF2]	Mixed
13					[0xBD, 0x01]	[0x5D, 0x01]	[0xF2, 0x01]	Mixed
14	0x01	0xBC	0x01	0xB0	[0xBD, 0x01]	[0x5D, 0x01]	[0xF3, 0x01]	Mixed
15					[0xBD, 0x01]	[0x5D, 0x01]	[0xF3, 0xF2]	Mixed
16					[0xBD, 0x01]	[0x5D, 0x5C]	[0xF2, 0x01]	Mixed
17					[0xBD, 0x01]	[0x5D, 0x5C]	[0xF3, 0x01]	Mixed
18					[0xBD, 0x01]	[0x5D, 0x5C]	[0xF3, 0xF2]	Mixed
19					[0xBD, 0xBC]	[0x5C, 0x01]	[0xF2, 0x01]	Mixed
20					[0xBD, 0xBC]	[0x5C, 0x01]	[0xF3, 0x01]	Mixed
21					[0xBD, 0xBC]	[0x5C, 0x01]	[0xF3, 0xF2]	Mixed
22					[0xBD, 0xBC]	[0x5D, 0x01]	[0xF2, 0x01]	Mixed
23					[0xBD, 0xBC]	[0x5D, 0x01]	[0xF3, 0x01]	Mixed
24					[0xBD, 0xBC]	[0x5D, 0x01]	[0xF3, 0xF2]	Mixed
25					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xF2, 0x01]	Mixed
26					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xF3, 0x01]	Mixed
27					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xF3, 0xF2]	All normal

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
28					[0xBC, 0x01]	[0x5C, 0x01]	[0xFE, 0x01]	All polynomial
29					[0xBC, 0x01]	[0x5C, 0x01]	[0xFF, 0x01]	Mixed
30					[0xBC, 0x01]	[0x5C, 0x01]	[0xFF, 0xFE]	Mixed
31					[0xBC, 0x01]	[0x5D, 0x01]	[0xFE, 0x01]	Mixed
32					[0xBC, 0x01]	[0x5D, 0x01]	[0xFF, 0x01]	Mixed
33					[0xBC, 0x01]	[0x5D, 0x01]	[0xFF, 0xFE]	Mixed
34					[0xBC, 0x01]	[0x5D, 0x5C]	[0xFE, 0x01]	Mixed
35					[0xBC, 0x01]	[0x5D, 0x5C]	[0xFF, 0x01]	Mixed
36					[0xBC, 0x01]	[0x5D, 0x5C]	[0xFF, 0xFE]	Mixed
37					[0xBD, 0x01]	[0x5C, 0x01]	[0xFE, 0x01]	Mixed
38					[0xBD, 0x01]	[0x5C, 0x01]	[0xFF, 0x01]	Mixed
39					[0xBD, 0x01]	[0x5C, 0x01]	[0xFF, 0xFE]	Mixed
40					[0xBD, 0x01]	[0x5D, 0x01]	[0xFE, 0x01]	Mixed
41	0x01	0xBC	0x01	0xEC	[0xBD, 0x01]	[0x5D, 0x01]	[0xFF, 0x01]	Mixed
42					[0xBD, 0x01]	[0x5D, 0x01]	[0xFF, 0xFE]	Mixed
43					[0xBD, 0x01]	[0x5D, 0x5C]	[0xFE, 0x01]	Mixed
44					[0xBD, 0x01]	[0x5D, 0x5C]	[0xFF, 0x01]	Mixed
45					[0xBD, 0x01]	[0x5D, 0x5C]	[0xFF, 0xFE]	Mixed
46					[0xBD, 0xBC]	[0x5C, 0x01]	[0xFE, 0x01]	Mixed
47					[0xBD, 0xBC]	[0x5C, 0x01]	[0xFF, 0x01]	Mixed
48					[0xBD, 0xBC]	[0x5C, 0x01]	[0xFF, 0xFE]	Mixed
49					[0xBD, 0xBC]	[0x5D, 0x01]	[0xFE, 0x01]	Mixed
50					[0xBD, 0xBC]	[0x5D, 0x01]	[0xFF, 0x01]	Mixed
51					[0xBD, 0xBC]	[0x5D, 0x01]	[0xFF, 0xFE]	Mixed
52					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xFE, 0x01]	Mixed
53					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xFF, 0x01]	Mixed
54					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xFF, 0xFE]	All normal
55					[0xBC, 0x01]	[0x5C, 0x01]	[0x42, 0x01]	All polynomial
56					[0xBC, 0x01]	[0x5C, 0x01]	[0x43, 0x01]	Mixed
57					[0xBC, 0x01]	[0x5C, 0x01]	[0x43, 0x42]	Mixed
58					[0xBC, 0x01]	[0x5D, 0x01]	[0x42, 0x01]	Mixed
59					[0xBC, 0x01]	[0x5D, 0x01]	[0x43, 0x01]	Mixed
60					[0xBC, 0x01]	[0x5D, 0x01]	[0x43, 0x42]	Mixed
61					[0xBC, 0x01]	[0x5D, 0x5C]	[0x42, 0x01]	Mixed
62					[0xBC, 0x01]	[0x5D, 0x5C]	[0x43, 0x01]	Mixed
63					[0xBC, 0x01]	[0x5D, 0x5C]	[0x43, 0x42]	Mixed
64					[0xBD, 0x01]	[0x5C, 0x01]	[0x42, 0x01]	Mixed
65					[0xBD, 0x01]	[0x5C, 0x01]	[0x43, 0x01]	Mixed
66					[0xBD, 0x01]	[0x5C, 0x01]	[0x43, 0x42]	Mixed
67					[0xBD, 0x01]	[0x5D, 0x01]	[0x42, 0x01]	Mixed
68	0x01	0xBC	0x01	0xED	[0xBD, 0x01]	[0x5D, 0x01]	[0x43, 0x01]	Mixed
69					[0xBD, 0x01]	[0x5D, 0x01]	[0x43, 0x42]	Mixed
70					[0xBD, 0x01]	[0x5D, 0x5C]	[0x42, 0x01]	Mixed
71					[0xBD, 0x01]	[0x5D, 0x5C]	[0x43, 0x01]	Mixed
72					[0xBD, 0x01]	[0x5D, 0x5C]	[0x43, 0x42]	Mixed
73					[0xBD, 0xBC]	[0x5C, 0x01]	[0x42, 0x01]	Mixed
74					[0xBD, 0xBC]	[0x5C, 0x01]	[0x43, 0x01]	Mixed
75					[0xBD, 0xBC]	[0x5C, 0x01]	[0x43, 0x42]	Mixed
76					[0xBD, 0xBC]	[0x5D, 0x01]	[0x42, 0x01]	Mixed
77					[0xBD, 0xBC]	[0x5D, 0x01]	[0x43, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
82					[0xBC, 0x01]	[0x5C, 0x01]	[0x4E, 0x01]	All polynomial
83					[0xBC, 0x01]	[0x5C, 0x01]	[0x4F, 0x01]	Mixed
84					[0xBC, 0x01]	[0x5C, 0x01]	[0x4F, 0x4E]	Mixed
85					[0xBC, 0x01]	[0x5D, 0x01]	[0x4E, 0x01]	Mixed
86					[0xBC, 0x01]	[0x5D, 0x01]	[0x4F, 0x01]	Mixed
87					[0xBC, 0x01]	[0x5D, 0x01]	[0x4F, 0x4E]	Mixed
88					[0xBC, 0x01]	[0x5D, 0x5C]	[0x4E, 0x01]	Mixed
89					[0xBC, 0x01]	[0x5D, 0x5C]	[0x4F, 0x01]	Mixed
90					[0xBC, 0x01]	[0x5D, 0x5C]	[0x4F, 0x4E]	Mixed
91					[0xBD, 0x01]	[0x5C, 0x01]	[0x4E, 0x01]	Mixed
92					[0xBD, 0x01]	[0x5C, 0x01]	[0x4F, 0x01]	Mixed
93					[0xBD, 0x01]	[0x5C, 0x01]	[0x4F, 0x4E]	Mixed
94					[0xBD, 0x01]	[0x5D, 0x01]	[0x4E, 0x01]	Mixed
95	0x01	0xBC	0x01	0xB1	[0xBD, 0x01]	[0x5D, 0x01]	[0x4F, 0x01]	Mixed
96					[0xBD, 0x01]	[0x5D, 0x01]	[0x4F, 0x4E]	Mixed
97					[0xBD, 0x01]	[0x5D, 0x5C]	[0x4E, 0x01]	Mixed
98					[0xBD, 0x01]	[0x5D, 0x5C]	[0x4F, 0x01]	Mixed
99					[0xBD, 0x01]	[0x5D, 0x5C]	[0x4F, 0x4E]	Mixed
100					[0xBD, 0xBC]	[0x5C, 0x01]	[0x4E, 0x01]	Mixed
101					[0xBD, 0xBC]	[0x5C, 0x01]	[0x4F, 0x01]	Mixed
102					[0xBD, 0xBC]	[0x5C, 0x01]	[0x4F, 0x4E]	Mixed
103					[0xBD, 0xBC]	[0x5D, 0x01]	[0x4E, 0x01]	Mixed
104					[0xBD, 0xBC]	[0x5D, 0x01]	[0x4F, 0x01]	Mixed
105					[0xBD, 0xBC]	[0x5D, 0x01]	[0x4F, 0x4E]	Mixed
106					[0xBD, 0xBC]	[0x5D, 0x5C]	[0x4E, 0x01]	Mixed
107					[0xBD, 0xBC]	[0x5D, 0x5C]	[0x4F, 0x01]	Mixed
108					[0xBD, 0xBC]	[0x5D, 0x5C]	[0x4F, 0x4E]	All normal
109					[0xBC, 0x01]	[0x5C, 0x01]	[0xAE, 0x01]	All polynomial
110					[0xBC, 0x01]	[0x5C, 0x01]	[0xAF, 0x01]	Mixed
111					[0xBC, 0x01]	[0x5C, 0x01]	[0xAF, 0xAE]	Mixed
112					[0xBC, 0x01]	[0x5D, 0x01]	[0xAE, 0x01]	Mixed
113					[0xBC, 0x01]	[0x5D, 0x01]	[0xAF, 0x01]	Mixed
114					[0xBC, 0x01]	[0x5D, 0x01]	[0xAF, 0xAE]	Mixed
115					[0xBC, 0x01]	[0x5D, 0x5C]	[0xAE, 0x01]	Mixed
116					[0xBC, 0x01]	[0x5D, 0x5C]	[0xAF, 0x01]	Mixed
117					[0xBC, 0x01]	[0x5D, 0x5C]	[0xAF, 0xAE]	Mixed
118					[0xBD, 0x01]	[0x5C, 0x01]	[0xAE, 0x01]	Mixed
119					[0xBD, 0x01]	[0x5C, 0x01]	[0xAF, 0x01]	Mixed
120					[0xBD, 0x01]	[0x5C, 0x01]	[0xAF, 0xAE]	Mixed
121					[0xBD, 0x01]	[0x5D, 0x01]	[0xAE, 0x01]	Mixed
122	0x01	0xBC	0x01	0x0C	[0xBD, 0x01]	[0x5D, 0x01]	[0xAF, 0x01]	Mixed
123					[0xBD, 0x01]	[0x5D, 0x01]	[0xAF, 0xAE]	Mixed
124					[0xBD, 0x01]	[0x5D, 0x5C]	[0xAE, 0x01]	Mixed
125					[0xBD, 0x01]	[0x5D, 0x5C]	[0xAF, 0x01]	Mixed
126					[0xBD, 0x01]	[0x5D, 0x5C]	[0xAF, 0xAE]	Mixed
127					[0xBD, 0xBC]	[0x5C, 0x01]	[0xAE, 0x01]	Mixed
128					[0xBD, 0xBC]	[0x5C, 0x01]	[0xAF, 0x01]	Mixed
129					[0xBD, 0xBC]	[0x5C, 0x01]	[0xAF, 0xAE]	Mixed
130					[0xBD, 0xBC]	[0x5D, 0x01]	[0xAE, 0x01]	Mixed
131					[0xBD, 0xBC]	[0x5D, 0x01]	[0xAF, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
136					[0xBC, 0x01]	[0x5C, 0x01]	[0xA2, 0x01]	All polynomial
137					[0xBC, 0x01]	[0x5C, 0x01]	[0xA3, 0x01]	Mixed
138					[0xBC, 0x01]	[0x5C, 0x01]	[0xA3, 0xA2]	Mixed
139					[0xBC, 0x01]	[0x5D, 0x01]	[0xA2, 0x01]	Mixed
140					[0xBC, 0x01]	[0x5D, 0x01]	[0xA3, 0x01]	Mixed
141					[0xBC, 0x01]	[0x5D, 0x01]	[0xA3, 0xA2]	Mixed
142					[0xBC, 0x01]	[0x5D, 0x5C]	[0xA2, 0x01]	Mixed
143					[0xBC, 0x01]	[0x5D, 0x5C]	[0xA3, 0x01]	Mixed
144					[0xBC, 0x01]	[0x5D, 0x5C]	[0xA3, 0xA2]	Mixed
145					[0xBD, 0x01]	[0x5C, 0x01]	[0xA2, 0x01]	Mixed
146					[0xBD, 0x01]	[0x5C, 0x01]	[0xA3, 0x01]	Mixed
147					[0xBD, 0x01]	[0x5C, 0x01]	[0xA3, 0xA2]	Mixed
148					[0xBD, 0x01]	[0x5D, 0x01]	[0xA2, 0x01]	Mixed
149	0x01	0xBC	0x01	0x50	[0xBD, 0x01]	[0x5D, 0x01]	[0xA3, 0x01]	Mixed
150					[0xBD, 0x01]	[0x5D, 0x01]	[0xA3, 0xA2]	Mixed
151					[0xBD, 0x01]	[0x5D, 0x5C]	[0xA2, 0x01]	Mixed
152					[0xBD, 0x01]	[0x5D, 0x5C]	[0xA3, 0x01]	Mixed
153					[0xBD, 0x01]	[0x5D, 0x5C]	[0xA3, 0xA2]	Mixed
154					[0xBD, 0xBC]	[0x5C, 0x01]	[0xA2, 0x01]	Mixed
155					[0xBD, 0xBC]	[0x5C, 0x01]	[0xA3, 0x01]	Mixed
156					[0xBD, 0xBC]	[0x5C, 0x01]	[0xA3, 0xA2]	Mixed
157					[0xBD, 0xBC]	[0x5D, 0x01]	[0xA2, 0x01]	Mixed
158					[0xBD, 0xBC]	[0x5D, 0x01]	[0xA3, 0x01]	Mixed
159					[0xBD, 0xBC]	[0x5D, 0x01]	[0xA3, 0xA2]	Mixed
160					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xA2, 0x01]	Mixed
161					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xA3, 0x01]	Mixed
162					[0xBD, 0xBC]	[0x5D, 0x5C]	[0xA3, 0xA2]	All normal
163					[0xBC, 0x01]	[0x5C, 0x01]	[0x1E, 0x01]	All polynomial
164					[0xBC, 0x01]	[0x5C, 0x01]	[0x1F, 0x01]	Mixed
165					[0xBC, 0x01]	[0x5C, 0x01]	[0x1F, 0x1E]	Mixed
166					[0xBC, 0x01]	[0x5D, 0x01]	[0x1E, 0x01]	Mixed
167					[0xBC, 0x01]	[0x5D, 0x01]	[0x1F, 0x01]	Mixed
168					[0xBC, 0x01]	[0x5D, 0x01]	[0x1F, 0x1E]	Mixed
169					[0xBC, 0x01]	[0x5D, 0x5C]	[0x1E, 0x01]	Mixed
170					[0xBC, 0x01]	[0x5D, 0x5C]	[0x1F, 0x01]	Mixed
171					[0xBC, 0x01]	[0x5D, 0x5C]	[0x1F, 0x1E]	Mixed
172					[0xBD, 0x01]	[0x5C, 0x01]	[0x1E, 0x01]	Mixed
173					[0xBD, 0x01]	[0x5C, 0x01]	[0x1F, 0x01]	Mixed
174					[0xBD, 0x01]	[0x5C, 0x01]	[0x1F, 0x1E]	Mixed
175					[0xBD, 0x01]	[0x5D, 0x01]	[0x1E, 0x01]	Mixed
176	0x01	0xBC	0x01	0x51	[0xBD, 0x01]	[0x5D, 0x01]	[0x1F, 0x01]	Mixed
177					[0xBD, 0x01]	[0x5D, 0x01]	[0x1F, 0x1E]	Mixed
178					[0xBD, 0x01]	[0x5D, 0x5C]	[0x1E, 0x01]	Mixed
179					[0xBD, 0x01]	[0x5D, 0x5C]	[0x1F, 0x01]	Mixed
180					[0xBD, 0x01]	[0x5D, 0x5C]	[0x1F, 0x1E]	Mixed
181					[0xBD, 0xBC]	[0x5C, 0x01]	[0x1E, 0x01]	Mixed
182					[0xBD, 0xBC]	[0x5C, 0x01]	[0x1F, 0x01]	Mixed
183					[0xBD, 0xBC]	[0x5C, 0x01]	[0x1F, 0x1E]	Mixed
184					[0xBD, 0xBC]	[0x5D, 0x01]	[0x1E, 0x01]	Mixed
185					[0xBD, 0xBC]	[0x5D, 0x01]	[0x1F, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
190					[0xBC, 0x01]	[0x5C, 0x01]	[0x12, 0x01]	All polynomial
191					[0xBC, 0x01]	[0x5C, 0x01]	[0x13, 0x01]	Mixed
192					[0xBC, 0x01]	[0x5C, 0x01]	[0x13, 0x12]	Mixed
193					[0xBC, 0x01]	[0x5D, 0x01]	[0x12, 0x01]	Mixed
194					[0xBC, 0x01]	[0x5D, 0x01]	[0x13, 0x01]	Mixed
195					[0xBC, 0x01]	[0x5D, 0x01]	[0x13, 0x12]	Mixed
196					[0xBC, 0x01]	[0x5D, 0x5C]	[0x12, 0x01]	Mixed
197					[0xBC, 0x01]	[0x5D, 0x5C]	[0x13, 0x01]	Mixed
198					[0xBC, 0x01]	[0x5D, 0x5C]	[0x13, 0x12]	Mixed
199					[0xBD, 0x01]	[0x5C, 0x01]	[0x12, 0x01]	Mixed
200					[0xBD, 0x01]	[0x5C, 0x01]	[0x13, 0x01]	Mixed
201					[0xBD, 0x01]	[0x5C, 0x01]	[0x13, 0x12]	Mixed
202					[0xBD, 0x01]	[0x5D, 0x01]	[0x12, 0x01]	Mixed
203	0x01	0xBC	0x01	0x0D	[0xBD, 0x01]	[0x5D, 0x01]	[0x13, 0x01]	Mixed
204					[0xBD, 0x01]	[0x5D, 0x01]	[0x13, 0x12]	Mixed
205					[0xBD, 0x01]	[0x5D, 0x5C]	[0x12, 0x01]	Mixed
206					[0xBD, 0x01]	[0x5D, 0x5C]	[0x13, 0x01]	Mixed
207					[0xBD, 0x01]	[0x5D, 0x5C]	[0x13, 0x12]	Mixed
208					[0xBD, 0xBC]	[0x5C, 0x01]	[0x12, 0x01]	Mixed
209					[0xBD, 0xBC]	[0x5C, 0x01]	[0x13, 0x01]	Mixed
210					[0xBD, 0xBC]	[0x5C, 0x01]	[0x13, 0x12]	Mixed
211					[0xBD, 0xBC]	[0x5D, 0x01]	[0x12, 0x01]	Mixed
212					[0xBD, 0xBC]	[0x5D, 0x01]	[0x13, 0x01]	Mixed
213					[0xBD, 0xBC]	[0x5D, 0x01]	[0x13, 0x12]	Mixed
214					[0xBD, 0xBC]	[0x5D, 0x5C]	[0x12, 0x01]	Mixed
215					[0xBD, 0xBC]	[0x5D, 0x5C]	[0x13, 0x01]	Mixed
216					[0xBD, 0xBC]	[0x5D, 0x5C]	[0x13, 0x12]	All normal
217					[0xBC, 0x01]	[0xE0, 0x01]	[0x12, 0x01]	All polynomial
218					[0xBC, 0x01]	[0xE0, 0x01]	[0x13, 0x01]	Mixed
219					[0xBC, 0x01]	[0xE0, 0x01]	[0x13, 0x12]	Mixed
220					[0xBC, 0x01]	[0xE1, 0x01]	[0x12, 0x01]	Mixed
221					[0xBC, 0x01]	[0xE1, 0x01]	[0x13, 0x01]	Mixed
222					[0xBC, 0x01]	[0xE1, 0x01]	[0x13, 0x12]	Mixed
223					[0xBC, 0x01]	[0xE1, 0xE0]	[0x12, 0x01]	Mixed
224					[0xBC, 0x01]	[0xE1, 0xE0]	[0x13, 0x01]	Mixed
225					[0xBC, 0x01]	[0xE1, 0xE0]	[0x13, 0x12]	Mixed
226					[0xBD, 0x01]	[0xE0, 0x01]	[0x12, 0x01]	Mixed
227					[0xBD, 0x01]	[0xE0, 0x01]	[0x13, 0x01]	Mixed
228					[0xBD, 0x01]	[0xE0, 0x01]	[0x13, 0x12]	Mixed
229					[0xBD, 0x01]	[0xE1, 0x01]	[0x12, 0x01]	Mixed
230	0x01	0xBD	0x01	0x0D	[0xBD, 0x01]	[0xE1, 0x01]	[0x13, 0x01]	Mixed
231					[0xBD, 0x01]	[0xE1, 0x01]	[0x13, 0x12]	Mixed
232					[0xBD, 0x01]	[0xE1, 0xE0]	[0x12, 0x01]	Mixed
233					[0xBD, 0x01]	[0xE1, 0xE0]	[0x13, 0x01]	Mixed
234					[0xBD, 0x01]	[0xE1, 0xE0]	[0x13, 0x12]	Mixed
235					[0xBD, 0xBC]	[0xE0, 0x01]	[0x12, 0x01]	Mixed
236					[0xBD, 0xBC]	[0xE0, 0x01]	[0x13, 0x01]	Mixed
237					[0xBD, 0xBC]	[0xE0, 0x01]	[0x13, 0x12]	Mixed
238					[0xBD, 0xBC]	[0xE1, 0x01]	[0x12, 0x01]	Mixed
239					[0xBD, 0xBC]	[0xE1, 0x01]	[0x13, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
244					[0xBC, 0x01]	[0xE0, 0x01]	[0x42, 0x01]	All polynomial
245					[0xBC, 0x01]	[0xE0, 0x01]	[0x43, 0x01]	Mixed
246					[0xBC, 0x01]	[0xE0, 0x01]	[0x43, 0x42]	Mixed
247					[0xBC, 0x01]	[0xE1, 0x01]	[0x42, 0x01]	Mixed
248					[0xBC, 0x01]	[0xE1, 0x01]	[0x43, 0x01]	Mixed
249					[0xBC, 0x01]	[0xE1, 0x01]	[0x43, 0x42]	Mixed
250					[0xBC, 0x01]	[0xE1, 0xE0]	[0x42, 0x01]	Mixed
251					[0xBC, 0x01]	[0xE1, 0xE0]	[0x43, 0x01]	Mixed
252					[0xBC, 0x01]	[0xE1, 0xE0]	[0x43, 0x42]	Mixed
253					[0xBD, 0x01]	[0xE0, 0x01]	[0x42, 0x01]	Mixed
254					[0xBD, 0x01]	[0xE0, 0x01]	[0x43, 0x01]	Mixed
255					[0xBD, 0x01]	[0xE0, 0x01]	[0x43, 0x42]	Mixed
256					[0xBD, 0x01]	[0xE1, 0x01]	[0x42, 0x01]	Mixed
257	0x01	0xBD	0x01	0xED	[0xBD, 0x01]	[0xE1, 0x01]	[0x43, 0x01]	Mixed
258					[0xBD, 0x01]	[0xE1, 0x01]	[0x43, 0x42]	Mixed
259					[0xBD, 0x01]	[0xE1, 0xE0]	[0x42, 0x01]	Mixed
260					[0xBD, 0x01]	[0xE1, 0xE0]	[0x43, 0x01]	Mixed
261					[0xBD, 0x01]	[0xE1, 0xE0]	[0x43, 0x42]	Mixed
262					[0xBD, 0xBC]	[0xE0, 0x01]	[0x42, 0x01]	Mixed
263					[0xBD, 0xBC]	[0xE0, 0x01]	[0x43, 0x01]	Mixed
264					[0xBD, 0xBC]	[0xE0, 0x01]	[0x43, 0x42]	Mixed
265					[0xBD, 0xBC]	[0xE1, 0x01]	[0x42, 0x01]	Mixed
266					[0xBD, 0xBC]	[0xE1, 0x01]	[0x43, 0x01]	Mixed
267					[0xBD, 0xBC]	[0xE1, 0x01]	[0x43, 0x42]	Mixed
268					[0xBD, 0xBC]	[0xE1, 0xE0]	[0x42, 0x01]	Mixed
269					[0xBD, 0xBC]	[0xE1, 0xE0]	[0x43, 0x01]	Mixed
270					[0xBD, 0xBC]	[0xE1, 0xE0]	[0x43, 0x42]	All normal
271					[0xBC, 0x01]	[0xE0, 0x01]	[0xFE, 0x01]	All polynomial
272					[0xBC, 0x01]	[0xE0, 0x01]	[0xFF, 0x01]	Mixed
273					[0xBC, 0x01]	[0xE0, 0x01]	[0xFF, 0xFE]	Mixed
274					[0xBC, 0x01]	[0xE1, 0x01]	[0xFE, 0x01]	Mixed
275					[0xBC, 0x01]	[0xE1, 0x01]	[0xFF, 0x01]	Mixed
276					[0xBC, 0x01]	[0xE1, 0x01]	[0xFF, 0xFE]	Mixed
277					[0xBC, 0x01]	[0xE1, 0xE0]	[0xFE, 0x01]	Mixed
278					[0xBC, 0x01]	[0xE1, 0xE0]	[0xFF, 0x01]	Mixed
279					[0xBC, 0x01]	[0xE1, 0xE0]	[0xFF, 0xFE]	Mixed
280					[0xBD, 0x01]	[0xE0, 0x01]	[0xFE, 0x01]	Mixed
281					[0xBD, 0x01]	[0xE0, 0x01]	[0xFF, 0x01]	Mixed
282					[0xBD, 0x01]	[0xE0, 0x01]	[0xFF, 0xFE]	Mixed
283					[0xBD, 0x01]	[0xE1, 0x01]	[0xFE, 0x01]	Mixed
284	0x01	0xBD	0x01	0xEC	[0xBD, 0x01]	[0xE1, 0x01]	[0xFF, 0x01]	Mixed
285					[0xBD, 0x01]	[0xE1, 0x01]	[0xFF, 0xFE]	Mixed
286					[0xBD, 0x01]	[0xE1, 0xE0]	[0xFE, 0x01]	Mixed
287					[0xBD, 0x01]	[0xE1, 0xE0]	[0xFF, 0x01]	Mixed
288					[0xBD, 0x01]	[0xE1, 0xE0]	[0xFF, 0xFE]	Mixed
289					[0xBD, 0xBC]	[0xE0, 0x01]	[0xFE, 0x01]	Mixed
290					[0xBD, 0xBC]	[0xE0, 0x01]	[0xFF, 0x01]	Mixed
291					[0xBD, 0xBC]	[0xE0, 0x01]	[0xFF, 0xFE]	Mixed
292					[0xBD, 0xBC]	[0xE1, 0x01]	[0xFE, 0x01]	Mixed
293					[0xBD, 0xBC]	[0xE1, 0x01]	[0xFF, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
298					[0xBC, 0x01]	[0xE0, 0x01]	[0xAE, 0x01]	All polynomial
299					[0xBC, 0x01]	[0xE0, 0x01]	[0xAF, 0x01]	Mixed
300					[0xBC, 0x01]	[0xE0, 0x01]	[0xAF, 0xAE]	Mixed
301					[0xBC, 0x01]	[0xE1, 0x01]	[0xAE, 0x01]	Mixed
302					[0xBC, 0x01]	[0xE1, 0x01]	[0xAF, 0x01]	Mixed
303					[0xBC, 0x01]	[0xE1, 0x01]	[0xAF, 0xAE]	Mixed
304					[0xBC, 0x01]	[0xE1, 0xE0]	[0xAE, 0x01]	Mixed
305					[0xBC, 0x01]	[0xE1, 0xE0]	[0xAF, 0x01]	Mixed
306					[0xBC, 0x01]	[0xE1, 0xE0]	[0xAF, 0xAE]	Mixed
307					[0xBD, 0x01]	[0xE0, 0x01]	[0xAE, 0x01]	Mixed
308					[0xBD, 0x01]	[0xE0, 0x01]	[0xAF, 0x01]	Mixed
309					[0xBD, 0x01]	[0xE0, 0x01]	[0xAF, 0xAE]	Mixed
310					[0xBD, 0x01]	[0xE1, 0x01]	[0xAE, 0x01]	Mixed
311	0x01	0xBD	0x01	0x0C	[0xBD, 0x01]	[0xE1, 0x01]	[0xAF, 0x01]	Mixed
312					[0xBD, 0x01]	[0xE1, 0x01]	[0xAF, 0xAE]	Mixed
313					[0xBD, 0x01]	[0xE1, 0xE0]	[0xAE, 0x01]	Mixed
314					[0xBD, 0x01]	[0xE1, 0xE0]	[0xAF, 0x01]	Mixed
315					[0xBD, 0x01]	[0xE1, 0xE0]	[0xAF, 0xAE]	Mixed
316					[0xBD, 0xBC]	[0xE0, 0x01]	[0xAE, 0x01]	Mixed
317					[0xBD, 0xBC]	[0xE0, 0x01]	[0xAF, 0x01]	Mixed
318					[0xBD, 0xBC]	[0xE0, 0x01]	[0xAF, 0xAE]	Mixed
319					[0xBD, 0xBC]	[0xE1, 0x01]	[0xAE, 0x01]	Mixed
320					[0xBD, 0xBC]	[0xE1, 0x01]	[0xAF, 0x01]	Mixed
321					[0xBD, 0xBC]	[0xE1, 0x01]	[0xAF, 0xAE]	Mixed
322					[0xBD, 0xBC]	[0xE1, 0xE0]	[0xAE, 0x01]	Mixed
323					[0xBD, 0xBC]	[0xE1, 0xE0]	[0xAF, 0x01]	Mixed
324					[0xBD, 0xBC]	[0xE1, 0xE0]	[0xAF, 0xAE]	All normal
325					[0xBC, 0x01]	[0xE0, 0x01]	[0x4E, 0x01]	All polynomial
326					[0xBC, 0x01]	[0xE0, 0x01]	[0x4F, 0x01]	Mixed
327					[0xBC, 0x01]	[0xE0, 0x01]	[0x4F, 0x4E]	Mixed
328					[0xBC, 0x01]	[0xE1, 0x01]	[0x4E, 0x01]	Mixed
329					[0xBC, 0x01]	[0xE1, 0x01]	[0x4F, 0x01]	Mixed
330					[0xBC, 0x01]	[0xE1, 0x01]	[0x4F, 0x4E]	Mixed
331					[0xBC, 0x01]	[0xE1, 0xE0]	[0x4E, 0x01]	Mixed
332					[0xBC, 0x01]	[0xE1, 0xE0]	[0x4F, 0x01]	Mixed
333					[0xBC, 0x01]	[0xE1, 0xE0]	[0x4F, 0x4E]	Mixed
334					[0xBD, 0x01]	[0xE0, 0x01]	[0x4E, 0x01]	Mixed
335					[0xBD, 0x01]	[0xE0, 0x01]	[0x4F, 0x01]	Mixed
336					[0xBD, 0x01]	[0xE0, 0x01]	[0x4F, 0x4E]	Mixed
337					[0xBD, 0x01]	[0xE1, 0x01]	[0x4E, 0x01]	Mixed
338	0x01	0xBD	0x01	0xB1	[0xBD, 0x01]	[0xE1, 0x01]	[0x4F, 0x01]	Mixed
339					[0xBD, 0x01]	[0xE1, 0x01]	[0x4F, 0x4E]	Mixed
340					[0xBD, 0x01]	[0xE1, 0xE0]	[0x4E, 0x01]	Mixed
341					[0xBD, 0x01]	[0xE1, 0xE0]	[0x4F, 0x01]	Mixed
342					[0xBD, 0x01]	[0xE1, 0xE0]	[0x4F, 0x4E]	Mixed
343					[0xBD, 0xBC]	[0xE0, 0x01]	[0x4E, 0x01]	Mixed
344					[0xBD, 0xBC]	[0xE0, 0x01]	[0x4F, 0x01]	Mixed
345					[0xBD, 0xBC]	[0xE0, 0x01]	[0x4F, 0x4E]	Mixed
346					[0xBD, 0xBC]	[0xE1, 0x01]	[0x4E, 0x01]	Mixed
347					[0xBD, 0xBC]	[0xE1, 0x01]	[0x4F, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
352					[0xBC, 0x01]	[0xE0, 0x01]	[0x1E, 0x01]	All polynomial
353					[0xBC, 0x01]	[0xE0, 0x01]	[0x1F, 0x01]	Mixed
354					[0xBC, 0x01]	[0xE0, 0x01]	[0x1F, 0x1E]	Mixed
355					[0xBC, 0x01]	[0xE1, 0x01]	[0x1E, 0x01]	Mixed
356					[0xBC, 0x01]	[0xE1, 0x01]	[0x1F, 0x01]	Mixed
357					[0xBC, 0x01]	[0xE1, 0x01]	[0x1F, 0x1E]	Mixed
358					[0xBC, 0x01]	[0xE1, 0xE0]	[0x1E, 0x01]	Mixed
359					[0xBC, 0x01]	[0xE1, 0xE0]	[0x1F, 0x01]	Mixed
360					[0xBC, 0x01]	[0xE1, 0xE0]	[0x1F, 0x1E]	Mixed
361					[0xBD, 0x01]	[0xE0, 0x01]	[0x1E, 0x01]	Mixed
362					[0xBD, 0x01]	[0xE0, 0x01]	[0x1F, 0x01]	Mixed
363					[0xBD, 0x01]	[0xE0, 0x01]	[0x1F, 0x1E]	Mixed
364					[0xBD, 0x01]	[0xE1, 0x01]	[0x1E, 0x01]	Mixed
365	0x01	0xBD	0x01	0x51	[0xBD, 0x01]	[0xE1, 0x01]	[0x1F, 0x01]	Mixed
366					[0xBD, 0x01]	[0xE1, 0x01]	[0x1F, 0x1E]	Mixed
367					[0xBD, 0x01]	[0xE1, 0xE0]	[0x1E, 0x01]	Mixed
368					[0xBD, 0x01]	[0xE1, 0xE0]	[0x1F, 0x01]	Mixed
369					[0xBD, 0x01]	[0xE1, 0xE0]	[0x1F, 0x1E]	Mixed
370					[0xBD, 0xBC]	[0xE0, 0x01]	[0x1E, 0x01]	Mixed
371					[0xBD, 0xBC]	[0xE0, 0x01]	[0x1F, 0x01]	Mixed
372					[0xBD, 0xBC]	[0xE0, 0x01]	[0x1F, 0x1E]	Mixed
373					[0xBD, 0xBC]	[0xE1, 0x01]	[0x1E, 0x01]	Mixed
374					[0xBD, 0xBC]	[0xE1, 0x01]	[0x1F, 0x01]	Mixed
375					[0xBD, 0xBC]	[0xE1, 0x01]	[0x1F, 0x1E]	Mixed
376					[0xBD, 0xBC]	[0xE1, 0xE0]	[0x1E, 0x01]	Mixed
377					[0xBD, 0xBC]	[0xE1, 0xE0]	[0x1F, 0x01]	Mixed
378					[0xBD, 0xBC]	[0xE1, 0xE0]	[0x1F, 0x1E]	All normal
379					[0xBC, 0x01]	[0xE0, 0x01]	[0xA2, 0x01]	All polynomial
380					[0xBC, 0x01]	[0xE0, 0x01]	[0xA3, 0x01]	Mixed
381					[0xBC, 0x01]	[0xE0, 0x01]	[0xA3, 0xA2]	Mixed
382					[0xBC, 0x01]	[0xE1, 0x01]	[0xA2, 0x01]	Mixed
383					[0xBC, 0x01]	[0xE1, 0x01]	[0xA3, 0x01]	Mixed
384					[0xBC, 0x01]	[0xE1, 0x01]	[0xA3, 0xA2]	Mixed
385					[0xBC, 0x01]	[0xE1, 0xE0]	[0xA2, 0x01]	Mixed
386					[0xBC, 0x01]	[0xE1, 0xE0]	[0xA3, 0x01]	Mixed
387					[0xBC, 0x01]	[0xE1, 0xE0]	[0xA3, 0xA2]	Mixed
388					[0xBD, 0x01]	[0xE0, 0x01]	[0xA2, 0x01]	Mixed
389					[0xBD, 0x01]	[0xE0, 0x01]	[0xA3, 0x01]	Mixed
390					[0xBD, 0x01]	[0xE0, 0x01]	[0xA3, 0xA2]	Mixed
391					[0xBD, 0x01]	[0xE1, 0x01]	[0xA2, 0x01]	Mixed
392	0x01	0xBD	0x01	0x50	[0xBD, 0x01]	[0xE1, 0x01]	[0xA3, 0x01]	Mixed
393					[0xBD, 0x01]	[0xE1, 0x01]	[0xA3, 0xA2]	Mixed
394					[0xBD, 0x01]	[0xE1, 0xE0]	[0xA2, 0x01]	Mixed
395					[0xBD, 0x01]	[0xE1, 0xE0]	[0xA3, 0x01]	Mixed
396					[0xBD, 0x01]	[0xE1, 0xE0]	[0xA3, 0xA2]	Mixed
397					[0xBD, 0xBC]	[0xE0, 0x01]	[0xA2, 0x01]	Mixed
398					[0xBD, 0xBC]	[0xE0, 0x01]	[0xA3, 0x01]	Mixed
399					[0xBD, 0xBC]	[0xE0, 0x01]	[0xA3, 0xA2]	Mixed
400					[0xBD, 0xBC]	[0xE1, 0x01]	[0xA2, 0x01]	Mixed
401					[0xBD, 0xBC]	[0xE1, 0x01]	[0xA3, 0x01]	Mixed

Case#	Parameters				Bases			Base Type
	T	N	τ	ν	$\mathbb{F}_{2^2}$ over $\mathbb{F}_2$	$\mathbb{F}_{2^4}$ over $\mathbb{F}_{2^2}$	$\mathbb{F}_{2^8}$ over $\mathbb{F}_{2^4}$	
406					[0xBC, 0x01]	[0xE0, 0x01]	[0xF2, 0x01]	All polynomial
407					[0xBC, 0x01]	[0xE0, 0x01]	[0xF3, 0x01]	Mixed
408					[0xBC, 0x01]	[0xE0, 0x01]	[0xF3, 0xF2]	Mixed
409					[0xBC, 0x01]	[0xE1, 0x01]	[0xF2, 0x01]	Mixed
410					[0xBC, 0x01]	[0xE1, 0x01]	[0xF3, 0x01]	Mixed
411					[0xBC, 0x01]	[0xE1, 0x01]	[0xF3, 0xF2]	Mixed
412					[0xBC, 0x01]	[0xE1, 0xE0]	[0xF2, 0x01]	Mixed
413					[0xBC, 0x01]	[0xE1, 0xE0]	[0xF3, 0x01]	Mixed
414					[0xBC, 0x01]	[0xE1, 0xE0]	[0xF3, 0xF2]	Mixed
415					[0xBD, 0x01]	[0xE0, 0x01]	[0xF2, 0x01]	Mixed
416					[0xBD, 0x01]	[0xE0, 0x01]	[0xF3, 0x01]	Mixed
417					[0xBD, 0x01]	[0xE0, 0x01]	[0xF3, 0xF2]	Mixed
418					[0xBD, 0x01]	[0xE1, 0x01]	[0xF2, 0x01]	Mixed
419	0x01	0xBD	0x01	0xB0	[0xBD, 0x01]	[0xE1, 0x01]	[0xF3, 0x01]	Mixed
420					[0xBD, 0x01]	[0xE1, 0x01]	[0xF3, 0xF2]	Mixed
421					[0xBD, 0x01]	[0xE1, 0xE0]	[0xF2, 0x01]	Mixed
422					[0xBD, 0x01]	[0xE1, 0xE0]	[0xF3, 0x01]	Mixed
423					[0xBD, 0x01]	[0xE1, 0xE0]	[0xF3, 0xF2]	Mixed
424					[0xBD, 0xBC]	[0xE0, 0x01]	[0xF2, 0x01]	Mixed
425					[0xBD, 0xBC]	[0xE0, 0x01]	[0xF3, 0x01]	Mixed
426					[0xBD, 0xBC]	[0xE0, 0x01]	[0xF3, 0xF2]	Mixed
427					[0xBD, 0xBC]	[0xE1, 0x01]	[0xF2, 0x01]	Mixed
428					[0xBD, 0xBC]	[0xE1, 0x01]	[0xF3, 0x01]	Mixed
429					[0xBD, 0xBC]	[0xE1, 0x01]	[0xF3, 0xF2]	Mixed
430					[0xBD, 0xBC]	[0xE1, 0xE0]	[0xF2, 0x01]	Mixed
431					[0xBD, 0xBC]	[0xE1, 0xE0]	[0xF3, 0x01]	Mixed
432					[0xBD, 0xBC]	[0xE1, 0xE0]	[0xF3, 0xF2]	All normal

Verilog Code for the AES S-box

```
/* The AES S-box */
module AES (b, Sb);
        input [7:0] b;
        output [7:0] Sb;
        wire [7:0] g;
        wire [9:0] m;
        wire [3:0] p, l;
        wire [17:0] e;

        Input M1(b, g, m);
        Top M2(g, m, p);
        Middle M3(p, l);
        Bottom M4(g, m, l, e);
        Output M5(e, Sb);
endmodule

/* Gates implemented as modules to prevent
 unintentional optimization of the DC */
module XOR (t, a, b);
        output t;
        input a, b;
        xor(t, a, b);
endmodule

module XNOR (t, a, b);
        output t;
        input a, b;
        xnor(t, a, b);
endmodule

module NAND (t, a, b);
        output t;
        input a, b;
        nand(t, a, b);
endmodule

module NOR (t, a, b);
        output t;
        input a, b;
        nor(t, a, b);
endmodule

/* input matrix */
module Input (b, g, m);
        input [7:0] b;
        output [7:0] g;
        output [9:0] m;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15, t16, t17, t18, t19;
```

```
        XOR m1(t1, b[7], b[4]);
        XOR m2(t2, b[3], b[1]);
        XOR m3(t3, b[2], t2);
        XOR m4(t4, b[6], t3);
        XOR m5(t5, b[0], t4);
        XOR m6(t6, b[5], t3);
        XOR m7(t7, t5, t6);
        XOR m8(t8, b[4], t7);
        XOR m9(t9, t1, t2);
        XOR m10(t10, t1, t8);
        XOR m11(t11, b[0], t9);
        XOR m12(t12, b[4], b[2]);
        XOR m13(t13, b[1], t7);
        XOR m14(t14, b[7], b[2]);
        XOR m15(t15, t13, t14);
        XOR m16(t16, b[7], b[1]);
        XOR m17(t17, t12, t16);
        XOR m18(t18, t6, t9);
        XOR m19(t19, t7, t11);

        assign g = {b[0], t11, t5, t7, t8, t15, t10, t13};
        assign m = {t9, t17, t4, t1, t19, t14, t18, t12, t6, t16};
endmodule

/* top part: GF(2^4) multiplier and GF(2^4) square-scaler */
module Top (g, m, p);
        input [7:0] g;
        input [9:0] m;
        output [3:0] p;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13,
             t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24;

        NAND m1(t1, g[6], g[2]);
        NAND m2(t2, m[9], m[8]);
        NAND m3(t3, g[7], g[3]);
        XOR m4(t13, t1, t2);
        XOR m5(t14, t3, t1);

        NAND m6(t4, m[7], m[6]);
        NOR m7(t5, m[7], m[6]);
        NAND m8(t6, m[3], m[2]);
        NOR m9(t7, m[3], m[2]);
        NAND m10(t8, m[5], m[4]);
        XOR m11(t15, t5, t6);
        XOR m12(t16, t8, t7);
        XOR m13(t17, t4, t6);
        XOR m14(t18, t8, t6);

        NAND m15(t9, g[4], g[0]);
        NOR m16(t10, m[1], m[0]);
        NAND m17(t11, g[5], g[1]);
        NOR m18(t12, g[4], g[0]);
```

```
        XOR m19(t19, t9, t10);
        XOR m20(t20, t11, t12);

        XOR m21(t21, t13, t15);
        XOR m22(t22, t14, t16);
        XOR m23(t23, t19, t17);
        XOR m24(t24, t20, t18);

        assign p[3] = t21;
        assign p[2] = t22;
        assign p[1] = t23;
        assign p[0] = t24;
endmodule

/* middle part: GF(2^4) inverse */
module Middle (p, l);
        input [3:0] p;
        output [3:0] l;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15;

        NAND m1(t1, p[3], p[0]);
        NOR m2(t2, t1, p[2]);
        NAND m3(t3, p[2], p[0]);
        XOR m4(t4, p[1], t3);
        NOR m5(t5, p[2], t4);
        NAND m6(t6, p[1], t4);
        NOR m7(t7, p[3], t4);
        NOR m8(t8, t7, t2);
        XNOR m9(t9, t5, t7);
        XNOR m10(t10, t9, p[3]);
        NAND m11(t11, t6, t8);
        NAND m12(t12, t8, p[1]);
        XNOR m13(t13, p[0], t12);
        NAND m14(t14, t1, p[2]);
        NAND m15(t15, t9, t14);

        assign l[3] = t13;
        assign l[2] = t11;
        assign l[1] = t15;
        assign l[0] = t10;
endmodule

/* bottom part: GF(2^4) multipliers */
module Bottom (g, m, l, e);
        input [7:0] g;
        input [9:0] m;
        input [3:0] l;
        output [17:0] e;
        wire k4, k3, k2, k1, k0;

        XOR m1(k4, l[3], l[2]);
```

```
        XOR m2(k3, l[3], l[1]);
        XOR m3(k2, l[2], l[0]);
        XOR m4(k1, k3, k2);
        XOR m5(k0, l[1], l[0]);

        NAND m6(e[17], g[2], l[2]);
        NAND m7(e[16], g[3], l[3]);
        NAND m8(e[15], m[8], k4);

        NAND m9(e[14], m[2], k1);
        NAND m10(e[13], m[4], k2);
        NAND m11(e[12], m[6], k3);

        NAND m12(e[11], g[0], l[0]);
        NAND m13(e[10], g[1], l[1]);
        NAND m14(e[9], m[0], k0);

        NAND m15(e[8], g[6], l[2]);
        NAND m16(e[7], g[7], l[3]);
        NAND m17(e[6], m[9], k4);

        NAND m18(e[5], m[3], k1);
        NAND m19(e[4], m[5], k2);
        NAND m20(e[3], m[7], k3);

        NAND m21(e[2], g[4], l[0]);
        NAND m22(e[1], g[5], l[1]);
        NAND m23(e[0], m[1], k0);
endmodule

/* output matrix */
module Output (e, Sb);
        input [17:0] e;
        output [7:0] Sb;
        wire E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1, E0;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15, t16, t17;

        XOR m1(E11, e[17], e[16]);
        XOR m2(E10, e[15], e[16]);
        XOR m3(E9, e[14], e[13]);
        XOR m4(E8, e[12], e[13]);
        XOR m5(E7, e[11], e[10]);
        XOR m6(E6, e[9], e[10]);
        XOR m7(E5, e[8], e[7]);
        XOR m8(E4, e[6], e[7]);
        XOR m9(E3, e[5], e[4]);
        XOR m10(E2, e[3], e[4]);
        XOR m11(E1, e[2], e[1]);
        XOR m12(E0, e[0], e[1]);

        XOR m13(t1, E2, E0);
```

```
    XOR m14(t2, E10, t1);
    XOR m15(t3, E8, t2);
    XOR m16(t4, E5, E1);
    XOR m17(t5, E5, E3);
    XOR m18(t6, E7, t5);
    XOR m19(t7, E8, E6);
    XOR m20(t8, E2, t3);
    XOR m21(t9, E4, t8);
    XOR m22(t10, t1, t5);
    XNOR m23(t11, E9, t6);
    XNOR m24(t12, t4, t7);
    XOR m25(t13, t3, t6);
    XOR m26(t14, E11, t13);
    XNOR m27(t15, t1, t9);
    XOR m28(t16, t10, t12);
    XOR m29(t17, t4, t9);

    assign Sb = {t3, t15, t11, t9, t17, t14, t16, t12};
endmodule
```

Verilog Code for the Camellia S-box

```
/* The Camellia S-box */
module Camellia (b, Sb);
        input [7:0] b;
        output [7:0] Sb;
        wire [7:0] g;
        wire [9:0] m;
        wire [3:0] p, l;
        wire [17:0] e;

        Input M1(b, g, m);
        Top M2(g, m, p);
        Middle M3(p, l);
        Bottom M4(g, m, l, e);
        Output M5(e, Sb);
endmodule

/* Gates implemented as modules to prevent
 unintentional optimization of the DC */
module XOR (t, a, b);
        output t;
        input a, b;
        xor(t, a, b);
endmodule

module XNOR (t, a, b);
        output t;
        input a, b;
        xnor(t, a, b);
endmodule

module NAND (t, a, b);
        output t;
        input a, b;
        nand(t, a, b);
endmodule

module NOR (t, a, b);
        output t;
        input a, b;
        nor(t, a, b);
endmodule

/* input matrix */
module Input (b, g, m);
        input [7:0] b;
        output [7:0] g;
        output [9:0] m;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15, t16, t17, t18, t19;
```

```
        XNOR m1(t1, b[6], b[3]);
        XNOR m2(t2, b[4], b[2]);
        XOR m3(t3, b[4], b[1]);
        XOR m4(t4, t1, t3);
        XOR m5(t5, t2, t4);
        XOR m6(t6, b[5], t2);
        XNOR m7(t7, b[0], t6);
        XOR m8(t8, b[7], b[0]);
        XOR m9(t9, t1, t8);
        XOR m10(t10, t5, t9);
        XOR m11(t11, b[1], t9);
        XNOR m12(t12, b[0], t11);
        XOR m13(t13, b[4], t11);
        XNOR m14(t14, b[2], t11);
        XOR m15(t15, b[6], b[5]);
        XNOR m16(t16, t9, t15);
        XOR m17(t17, t7, t16);
        XNOR m18(t18, b[1], t15);
        XOR m19(t19, t6, t18);

        assign g = {t5, t4, t10, t1, t16, t17, t11, t12};
        assign m = {t2, t7, t9, t18, t3, t19, t13, t6, t14, ~b[0]};
endmodule

/* top part: GF(2^4) multiplier and GF(2^4) square-scaler */
module Top (g, m, d);
        input [7:0] g;
        input [9:0] m;
        output [3:0] d;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13,
             t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24;

        NAND m1(t1, g[7], g[3]);
        NOR m2(t2, g[6], g[2]);
        NOR m3(t3, g[7], g[3]);
        NAND m4(t4, m[9], m[8]);
        XOR m5(t13, t1, t2);
        XOR m6(t14, t3, t4);

        NAND m7(t5, m[5], m[4]);
        NAND m8(t6, m[3], m[2]);
        NAND m9(t7, m[7], m[6]);
        NOR m10(t8, m[3], m[2]);
        NOR m11(t9, m[5], m[4]);
        XOR m12(t15, t5, t6);
        XOR m13(t16, t7, t5);
        XOR m14(t17, t5, t8);
        XOR m15(t18, t7, t9);

        NAND m16(t10, g[5], g[1]);
        NAND m17(t11, g[4], g[0]);
        NAND m18(t12, m[1], m[0]);
```

```
        XOR m19(t19, t10, t11);
        XOR m20(t20, t10, t12);

        XOR m21(t21, t13, t15);
        XOR m22(t22, t14, t16);
        XOR m23(t23, t19, t17);
        XOR m24(t24, t20, t18);

        assign d[3] = t21;
        assign d[2] = t22;
        assign d[1] = t23;
        assign d[0] = t24;
endmodule

/* middle part: GF(2^4) inverse */
module Middle (p, l);
        input [3:0] p;
        output [3:0] l;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15;

        NAND m1(t1, p[3], p[0]);
        NOR m2(t2, t1, p[2]);
        NAND m3(t3, p[2], p[0]);
        XOR m4(t4, p[1], t3);
        NOR m5(t5, p[2], t4);
        NAND m6(t6, p[1], t4);
        NOR m7(t7, p[3], t4);
        NOR m8(t8, t7, t2);
        XNOR m9(t9, t5, t7);
        XNOR m10(t10, t9, p[3]);
        NAND m11(t11, t6, t8);
        NAND m12(t12, t8, p[1]);
        XNOR m13(t13, p[0], t12);
        NAND m14(t14, t1, p[2]);
        NAND m15(t15, t9, t14);

        assign l[3] = t13;
        assign l[2] = t11;
        assign l[1] = t15;
        assign l[0] = t10;
endmodule

/* bottom part: GF(2^4) multipliers */
module Bottom (g, m, l, e);
        input [7:0] g;
        input [9:0] m;
        input [3:0] l;
        output [17:0] e;
        wire k4, k3, k2, k1, k0;

        XOR m1(k4, l[3], l[2]);
```

```
        XOR m2(k3, l[3], l[1]);
        XOR m3(k2, l[2], l[0]);
        XOR m4(k1, k3, k2);
        XOR m5(k0, l[1], l[0]);

        NAND m6(e[17], g[2], l[2]);
        NAND m7(e[16], g[3], l[3]);
        NAND m8(e[15], m[8], k4);

        NAND m9(e[14], m[2], k1);
        NAND m10(e[13], m[4], k2);
        NAND m11(e[12], m[6], k3);

        NAND m12(e[11], g[0], l[0]);
        NAND m13(e[10], g[1], l[1]);
        NAND m14(e[9], m[0], k0);

        NAND m15(e[8], g[6], l[2]);
        NAND m16(e[7], g[7], l[3]);
        NAND m17(e[6], m[9], k4);

        NAND m18(e[5], m[3], k1);
        NAND m19(e[4], m[5], k2);
        NAND m20(e[3], m[7], k3);

        NAND m21(e[2], g[4], l[0]);
        NAND m22(e[1], g[5], l[1]);
        NAND m23(e[0], m[1], k0);
endmodule

/* output matrix */
module Output (e, Sb);
        input [17:0] e;
        output [7:0] Sb;
        wire E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1, E0;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15, t16;

        XOR m1(E11, e[17], e[16]);
        XOR m2(E10, e[15], e[16]);
        XOR m3(E9, e[14], e[13]);
        XOR m4(E8, e[12], e[13]);
        XOR m5(E7, e[11], e[10]);
        XOR m6(E6, e[9], e[10]);
        XOR m7(E5, e[8], e[7]);
        XOR m8(E4, e[6], e[7]);
        XOR m9(E3, e[5], e[4]);
        XOR m10(E2, e[3], e[4]);
        XOR m11(E1, e[2], e[1]);
        XOR m12(E0, e[0], e[1]);

        XNOR m13(t1, E2, E0);
```

```
        XNOR m14(t2, E10, t1);
        XOR m15(t3, E6, t2);
        XOR m16(t4, E11, t3);
        XOR m17(t5, E9, t4);
        XOR m18(t6, E11, E7);
        XOR m19(t7, E5, t6);
        XNOR m20(t8, E4, E0);
        XNOR m21(t9, t5, t8);
        XOR m22(t10, t2, t9);
        XNOR m23(t11, E1, t1);
        XOR m24(t12, t7, t9);
        XNOR m25(t13, E5, t11);
        XNOR m26(t14, E8, t10);
        XNOR m27(t15, E3, t12);
        XOR m28(t16, t7, t11);

        assign Sb = {t3, t1, t15, t5, t13, t14, t8, t16};
endmodule
```

Verilog Code for the SM4 S-box

```
/* The SM4 S-box */
module SM4 (b, Sb);
        input [7:0] b;
        output [7:0] Sb;
        wire [7:0] g;
        wire [9:0] m;
        wire [3:0] p, l;
        wire [17:0] e;

        Input M1(b, g, m);
        Top M2(g, m, p);
        Middle M3(p, l);
        Bottom M4(g, m, l, e);
        Output M5(e, Sb);
endmodule

/* Gates implemented as modules to prevent
 unintentional optimization of the DC */
module XOR (t, a, b);
        output t;
        input a, b;
        xor(t, a, b);
endmodule

module XNOR (t, a, b);
        output t;
        input a, b;
        xnor(t, a, b);
endmodule

module NAND (t, a, b);
        output t;
        input a, b;
        nand(t, a, b);
endmodule

module NOR (t, a, b);
        output t;
        input a, b;
        nor(t, a, b);
endmodule

/* input matrix */
module Input (b, g, m);
        input [7:0] b;
        output [7:0] g;
        output [9:0] m;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15, t16, t17;
```

```
        XOR m1(t1, b[7], b[5]);
        XNOR m2(t2, b[5], b[1]);
        XNOR m3(t3, b[0], t2);
        XOR m4(t4, b[6], b[2]);
        XOR m5(t5, b[3], t3);
        XOR m6(t6, b[4], t1);
        XOR m7(t7, b[1], t5);
        XOR m8(t8, b[1], t4);
        XOR m9(t9, t6, t8);
        XOR m10(t10, t6, t7);
        XNOR m11(t11, b[3], t1);
        XNOR m12(t12, b[6], t9);
        XOR m13(t13, t4, t10);
        XOR m14(t14, t2, t11);
        XOR m15(t15, t12, t14);
        XOR m16(t16, t3, t12);
        XOR m17(t17, t11, t16);

        assign g = {t15, t14, ~b[0], t2, t5, t13, t7, t10};
        assign m = {t12, t9, t17, b[1], t11, t4, t16, t8, t3, t6};
endmodule

/* top part: GF(2^4) multiplier and GF(2^4) square-scaler */
module Top (g, m, d);
        input [7:0] g;
        input [9:0] m;
        output [3:0] d;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13,
             t14, t15, t16, t17, t18, t19, t20, t21, t22, t23, t24;

        NAND m1(t1, g[5], g[1]);
        NAND m2(t2, m[1], m[0]);
        NAND m3(t3, g[4], g[0]);
        NAND m4(t4, g[7], g[3]);
        NAND m5(t5, m[9], m[8]);
        NOR m6(t6, g[6], g[2]);
        NOR m7(t7, g[7], g[3]);
        NOR m8(t8, m[9], m[8]);
        NOR m9(t9, m[7], m[6]);
        NAND m10(t10, m[3], m[2]);
        NAND m11(t11, m[5], m[4]);
        NOR m12(t12, m[3], m[2]);

        XOR m13(t13, t1, t2);
        XOR m14(t14, t3, t2);
        XOR m15(t15, t4, t13);
        XOR m16(t16, t5, t14);
        XOR m17(t17, t9, t10);
        XOR m18(t18, t11, t12);
        XOR m19(t19, t6, t15);
        XOR m20(t20, t7, t16);
```

```
        XOR m21(t21, t19, t17);
        XOR m22(t22, t20, t18);
        XOR m23(t23, t8, t15);
        XOR m24(t24, t6, t16);

        assign d[3] = t21;
        assign d[2] = t22;
        assign d[1] = t23;
        assign d[0] = t24;
endmodule

/* middle part: GF(2^4) inverse */
module Middle (p, l);
        input [3:0] p;
        output [3:0] l;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15;

        NAND m1(t1, p[3], p[0]);
        NOR m2(t2, t1, p[2]);
        NAND m3(t3, p[2], p[0]);
        XOR m4(t4, p[1], t3);
        NOR m5(t5, p[2], t4);
        NAND m6(t6, p[1], t4);
        NOR m7(t7, p[3], t4);
        NOR m8(t8, t7, t2);
        XNOR m9(t9, t5, t7);
        XNOR m10(t10, t9, p[3]);
        NAND m11(t11, t6, t8);
        NAND m12(t12, t8, p[1]);
        XNOR m13(t13, p[0], t12);
        NAND m14(t14, t1, p[2]);
        NAND m15(t15, t9, t14);

        assign l[3] = t13;
        assign l[2] = t11;
        assign l[1] = t15;
        assign l[0] = t10;
endmodule

/* bottom part: GF(2^4) multipliers */
module Bottom (g, m, l, e);
        input [7:0] g;
        input [9:0] m;
        input [3:0] l;
        output [17:0] e;
        wire k4, k3, k2, k1, k0;

        XOR m1(k4, l[3], l[2]);
        XOR m2(k3, l[3], l[1]);
        XOR m3(k2, l[2], l[0]);
        XOR m4(k1, k3, k2);
```

```
        XOR m5(k0, l[1], l[0]);

        NAND m6(e[17], g[2], l[2]);
        NAND m7(e[16], g[3], l[3]);
        NAND m8(e[15], m[8], k4);

        NAND m9(e[14], m[2], k1);
        NAND m10(e[13], m[4], k2);
        NAND m11(e[12], m[6], k3);

        NAND m12(e[11], g[0], l[0]);
        NAND m13(e[10], g[1], l[1]);
        NAND m14(e[9], m[0], k0);

        NAND m15(e[8], g[6], l[2]);
        NAND m16(e[7], g[7], l[3]);
        NAND m17(e[6], m[9], k4);

        NAND m18(e[5], m[3], k1);
        NAND m19(e[4], m[5], k2);
        NAND m20(e[3], m[7], k3);

        NAND m21(e[2], g[4], l[0]);
        NAND m22(e[1], g[5], l[1]);
        NAND m23(e[0], m[1], k0);
endmodule

/* output matrix */
module Output (e, Sb);
        input [17:0] e;
        output [7:0] Sb;
        wire E11, E10, E9, E8, E7, E6, E5, E4, E3, E2, E1, E0;
        wire t1, t2, t3, t4, t5, t6, t7, t8, t9, t10,
             t11, t12, t13, t14, t15, t16;

        XOR m1(E11, e[17], e[16]);
        XOR m2(E10, e[15], e[16]);
        XOR m3(E9, e[14], e[13]);
        XOR m4(E8, e[12], e[13]);
        XOR m5(E7, e[11], e[10]);
        XOR m6(E6, e[9], e[10]);
        XOR m7(E5, e[8], e[7]);
        XOR m8(E4, e[6], e[7]);
        XOR m9(E3, e[5], e[4]);
        XOR m10(E2, e[3], e[4]);
        XOR m11(E1, e[2], e[1]);
        XOR m12(E0, e[0], e[1]);

        XOR m13(t1, E9, E7);
        XOR m14(t2, E1, t1);
        XOR m15(t3, E3, t2);
        XOR m16(t4, E5, E3);
```

```
    XOR m17(t5, E4, t4);
    XOR m18(t6, E4, E0);
    XOR m19(t7, E11, E7);
    XOR m20(t8, t1, t4);
    XOR m21(t9, t1, t6);
    XOR m22(t10, E2, t5);
    XOR m23(t11, E10, E8);
    XNOR m24(t12, t3, t11);
    XOR m25(t13, t10, t12);
    XNOR m26(t14, t3, t7);
    XNOR m27(t15, E10, E6);
    XOR m28(t16, t6, t14);

    assign Sb = {t15, t13, t8, t14, t11, t9, t12, t16};
endmodule
```