HUNAN UNIVERSITY
湖南大学

功耗攻击防御技术在分组密码中的应用研究

硕士论文答辩

姓　　名：袁征
研究方向：嵌入式系统
指导教师：李仁发 教授

背 景 1/2

背景

介绍

本文工作

实验&结论

下一步工作

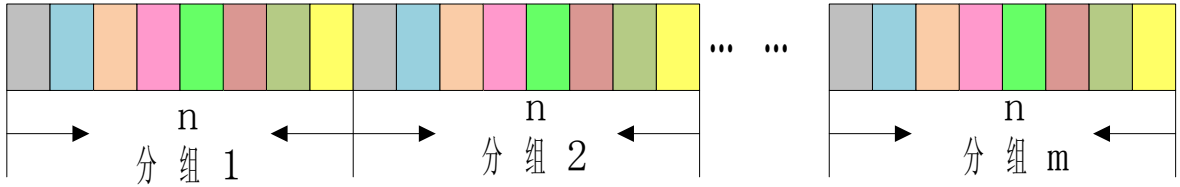
随着科学技术的不断发展，基于硬件的密码设备的应用越来越广泛，如汽车电子、遥控无钥门禁系统、智能卡、PDA、移动电话等。因而采取各种方式和手段确保硬件设备的安全性对于整个系统来说非常重要。

- 研究背景

- 实验结果及结论

分组密码以其明文信息良好的扩展性，对插入的敏感性，不需要密钥同步，较强的适用性和易于标准化等优点在信息安全领域有着越来越广泛的作用。

- 下一步工作



背 景 2/2

背景

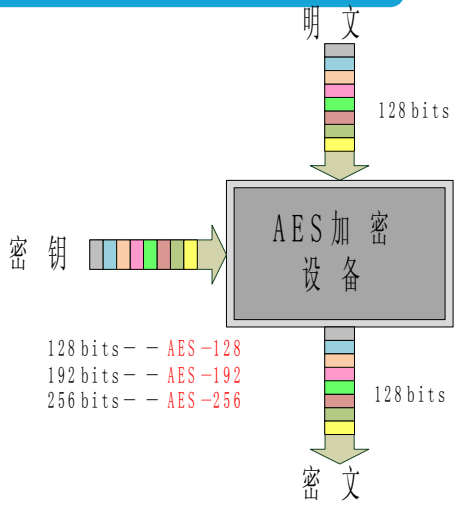
介绍

本文工作

实验&结论

下一步工作

*Rijndael*算法以其强安全性、高性能、易用性和灵活性等优点在美国国家标准与技术研究院(NIST)的公开评选中获胜，成为新的数据加密标准AES (*Advanced Encryption Standard*)。



- DPA、HO-DPA、glitch攻击等功耗攻击对AES造成了严重的威胁。
- 现有的防御技术在一定程度上存在着计算复杂、占用芯片面积大、速度慢、吞吐量低、无法抵抗HO-DPA攻击和glitch攻击等缺陷。

介绍 1/3 ——高级加密标准AES

背景

介绍

本文工作

实验&结论

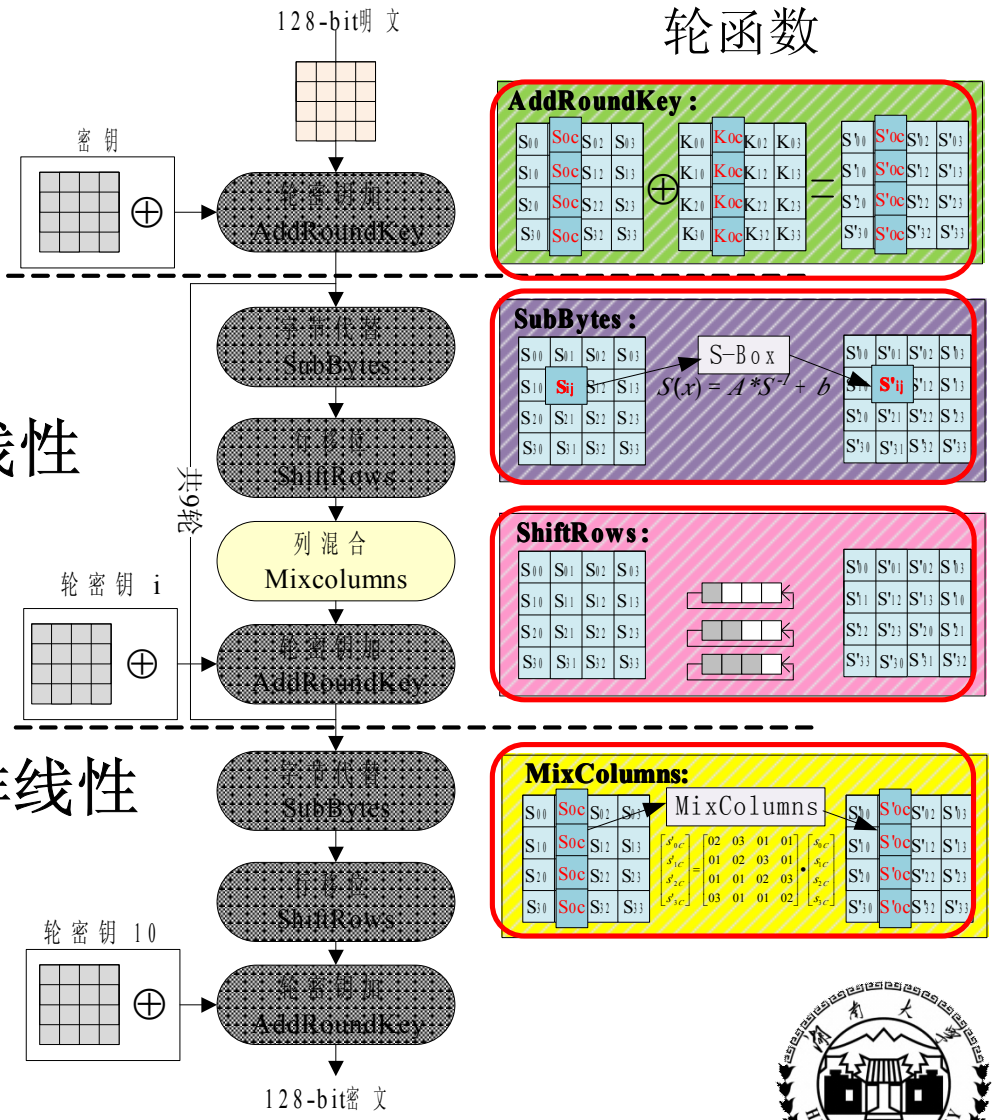
下一步工作

- AddRoundKey
- ShiftRows
- MixColumns

线性

- SubBytes (*S*-box)

非线性



介绍 2/3 ——功耗攻击技术

背景

介绍

本文工作

实验&结论

下一步工作

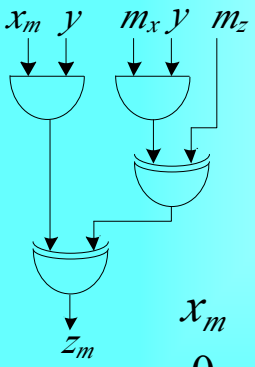
- 简单功耗攻击
- 差分功耗攻击
- 高阶差分功耗攻击
- 脉冲攻击

SPA是指利用功率轨迹的密钥

DPA记录了密码设备加密/解密

HO-DPA是指利用密码设备中

由于电路输入信号的到达时间不同会引起输出产生临时状态，**glitch**攻击就是通过对这些临时状态的攻击来获取电路信息。



$$z = x \text{ AND } y$$

$$x_m = x \oplus m_x$$

$$z_m = x_m \bullet y \bullet (m_x \bullet y \oplus m_z)$$

x_m	m_x	AND	XOR
0	0	0	0
0	1	1	2
1	0	1	1
1	1	2	2

介绍 3/3 ——功耗攻击防御技术

背景

介绍

本文工作

实验&结论

下一步工作

- 隐藏技术
- 乱序技术
- 掩码技术
- 秘密共享技术

Hiding是指通过将设备在每个时钟周

Shuffling是指改变密码算法的加密步

Masking是指将加密过程中的每一个

Secret Sharing是指将秘密信息按照一定方式拆分成*n*个份额，然后将这些份额分别分给不同的参与者，使每个参与者都拥有其中的一个share。在秘密信息恢复时，当且仅当足够数量的share结合起来，才能重建秘密信息。秘密共享近年来才应用于硬件加密。

本文工作

背景

介绍

本文工作

实验&结论

下一步工作

- 提出了一种$GF(2^4)$域上基于掩码的AES抗功耗攻击方案。
- 提出了一种基于秘密共享的AES抗功耗攻击方案。

本文工作

背景

介绍

本文工作

实验&结论

下一步工作

$a = [a_7,a_6,a_5,a_4,a_3,a_2,a_1,a_0]$　　$b = [b_7,b_6,b_5,b_4,b_3,b_2,b_1,b_0]$

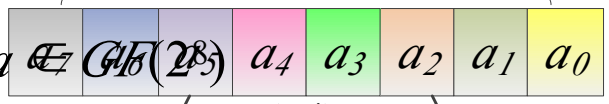
$GF(2^8)$:

- $a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$
- $b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$
- **$a(x)+b(x)$;　　$a(x)\cdot b(x)$;　　$b^{-1}(x)$**

- **$GF(2^8) \rightarrow GF(2^4)$**：$T_{map}$算法

T_{map}算法　$a = [a_7,a_6,a_5,a_4,a_3,a_2,a_1,a_0]$

$GF(2^8)$:



输入：$a \in GF(2^8)$

输出：a_h，$a_l \in GF(2^4)$

$GF(2^4)$:

1. 已知：$a = \{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\} \in GF(2^8)$
2. 计算：$a_A = a_1 + a_7$，　$a_B = a_5 + a_7$，　$a_C = a_4 + a_6$;
3. 令 $a_h = [a_{h3}, a_{h2}, a_{h1}, a_{h0}]$　$a_{h2} = a_B$　$a_{h3} = a_B + a_C + a_5$，　$a_{h1} = a_A + a_C$，　$a_{h0} = a_C + a_5$，

   $a_B = a_2 + a_4$，　$a_{l2} = a_A$，　$a_{l1} = a_1 + a_2$，　$a_{l0} = a_C + a_0 + a_5$
4. 输出：$a_h = \{a_{h3}, a_{h2}, a_{h1}, a_{h0}\}$，　$a_l = \{a_{l3}, a_{l2}, a_{l1}, a_{l0}\}$

本文工作 1/2 ——基于掩码的AES抗功耗攻击方案

背景

介绍

本文工作

实验&结论

下一步工作

- 在使用布尔掩码操作时，中间值x通过异或操作被掩码m掩盖，得到掩盖后的中间值$x_m = x + m$。

$ShiftRows(x+m) = ShiftRows(x) + ShiftRows(m)$

$MixColumn(x+m) = MixColumn(x) + MixColumn(m)$

$AddroundKey(x+m) = AddroundKey(x) + AddroundKey(m)$

$Sbox(x+m) \neq Sbox(x) + Sbox(m) \rightarrow$

$\boldsymbol{S'box}(x+m) = Sbox(x) + Sbox(m) = Sbox(x) + m'$

S-盒设计方案：

- $S(x) = A \cdot x^{-1} + B$

$$(a+m)^{-1} = ((a_h + m_h)x + (a_l + m_l))^{-1}$$
$$= (a'_h + m'_h)x + (a'_l + m'_l)$$
$$a_h' = a_h \times ((a_h^2 \times \{e\}) + (a_h \times a_l) + a_l^2)^{-1}$$
$$a_l' = (a_h + a_l) \times ((a_h^2 \times \{e\}) + (a_h \times a_l) + a_l^2)^{-1}$$

本文工作 1/2 ——基于掩码的AES抗功耗攻击方案

背景

介绍

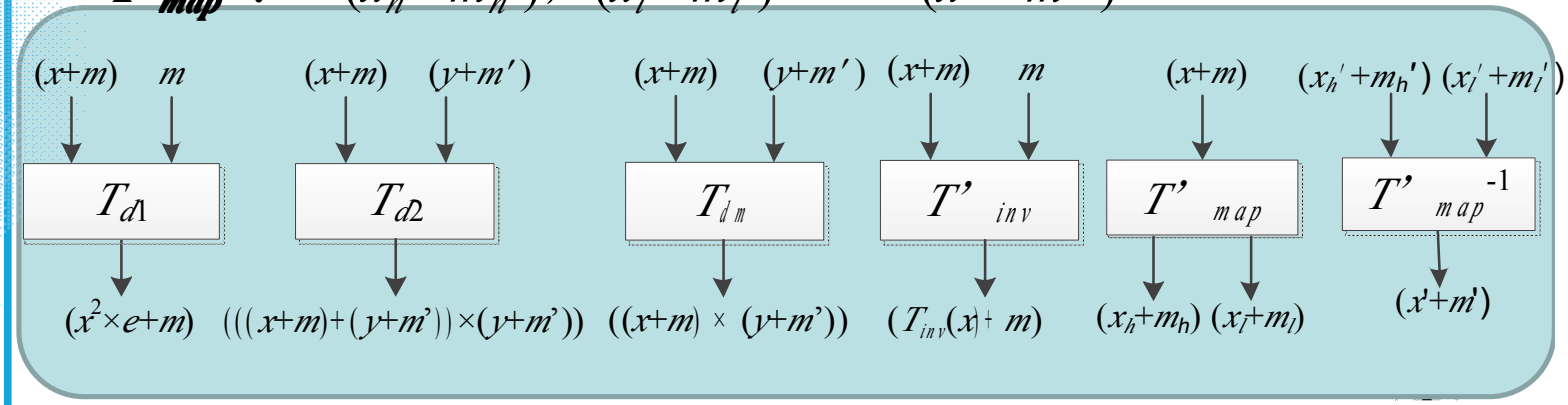
本文工作

实验&结论

下一步工作

- **LUT**：提前计算所有输入对应的输出值并存储起来，在算法执行时以“查表”的方式获得输出值。

LUT	输入	输出
T_{d1}:	$((x+m), m)$	$\rightarrow x^2 \times \{e\} + m$
T_{d2}:	$((x+m), (y+m'))$	$\rightarrow ((x+m) + (y+m')) \times (y+m')$
T_{dm}:	$((x+m), (y+m'))$	$\rightarrow (x+m) \times (y+m')$
T'_{inv}:	$((x+m), m)$	$\rightarrow T_{inv}(x) + m$
T'_{map}:	$(x+m)$	$\rightarrow [\,(x_h+m_h)\,,\ (x_l+m_l)\,]$
T'^{-1}_{map}:	$(x_h'+m_h'),\ (x_l'+m_l')$	$\rightarrow (x''+m'')$



本文工作 1/2 ——基于掩码的AES抗功耗攻击方案

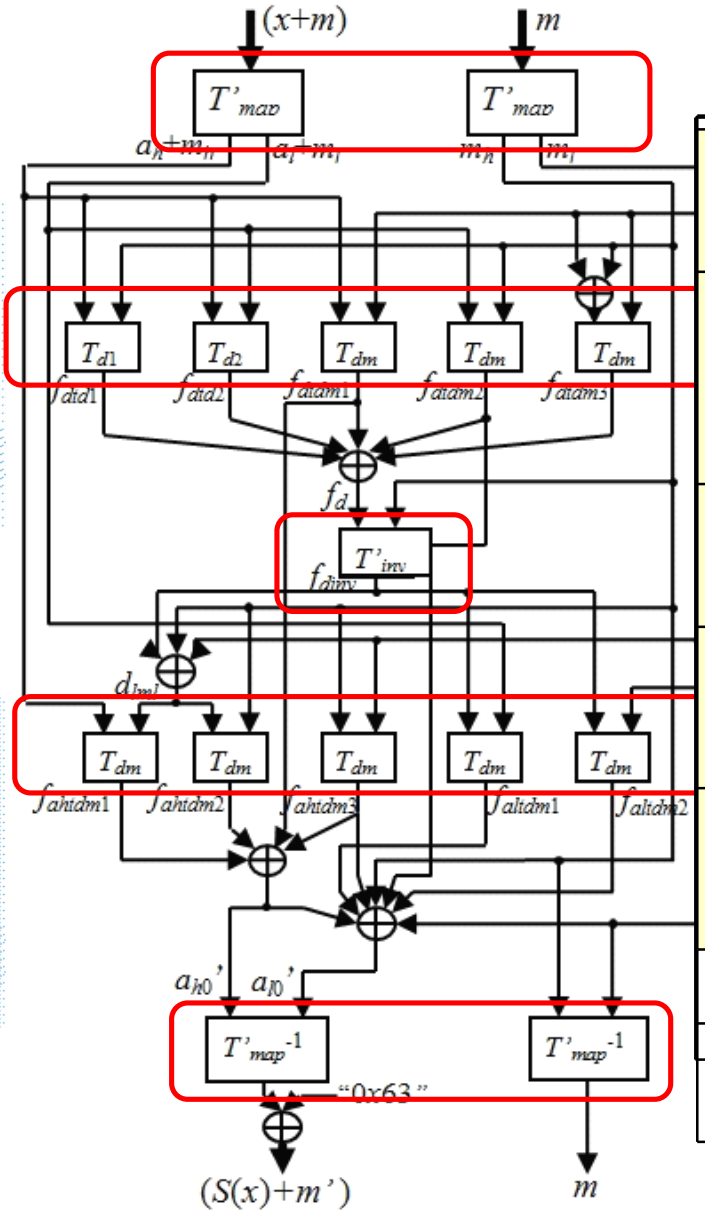
背景

介绍

本文工作

实验&结论

下一步工作



正确性：

a_{h0}'	$= f_{ahtdm1} + f_{ahtdm2} + f_{dtdm1} + f_{ahtdm3} + m_h$ $= a_h \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m_h$
a_{l0}'	$= a_{h0}' + f_{altdm1} + f_{ahtdm3} + m_h + f_{dtdm2} + f_{ahtdm2} + m_l$ $= a_h \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + a_l \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m_l$
原始的中间值：	
a_{h0}'	$= a_h \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m_h + m_h$ $= a_h \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1}$
a_{l0}'	$= (a_h \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + a_l \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m_l + m_l$ $= (a_h + a_l) \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1}$
	$= m_l \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m_h \times (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m_h$
d_{iml}	$= f_{dinv} + m_h + m_l$ $= (a_h^2 \times e + a_h \times a_l + a_l^2)^{-1} + m$

本文工作 1/2 ——基于掩码的AES抗功耗攻击方案

背景

介绍

本文工作

实验&结论

下一步工作

安全性：

- **抗HO-DPA攻击：**
  - *S*-盒内部的所有的中间值都被掩码掩盖；
  - 掩盖*S*-盒内部每个中间值的掩码并不相同；
  - 不能通过这些掩盖后的中间值获取信息。
- **抗glitch攻击：**
  - 新的*S*-盒(*S' box*)使用LUT查找表操作取代了易受glitch攻击的逻辑与门运算；
  - 所有中间值都被不同的掩码值掩盖；
  - glitch攻击难以成功。

本文工作 1/2 ——基于掩码的AES抗功耗攻击方案

背景

介绍

本文工作

实验&结论

下一步工作

AES设计方案：

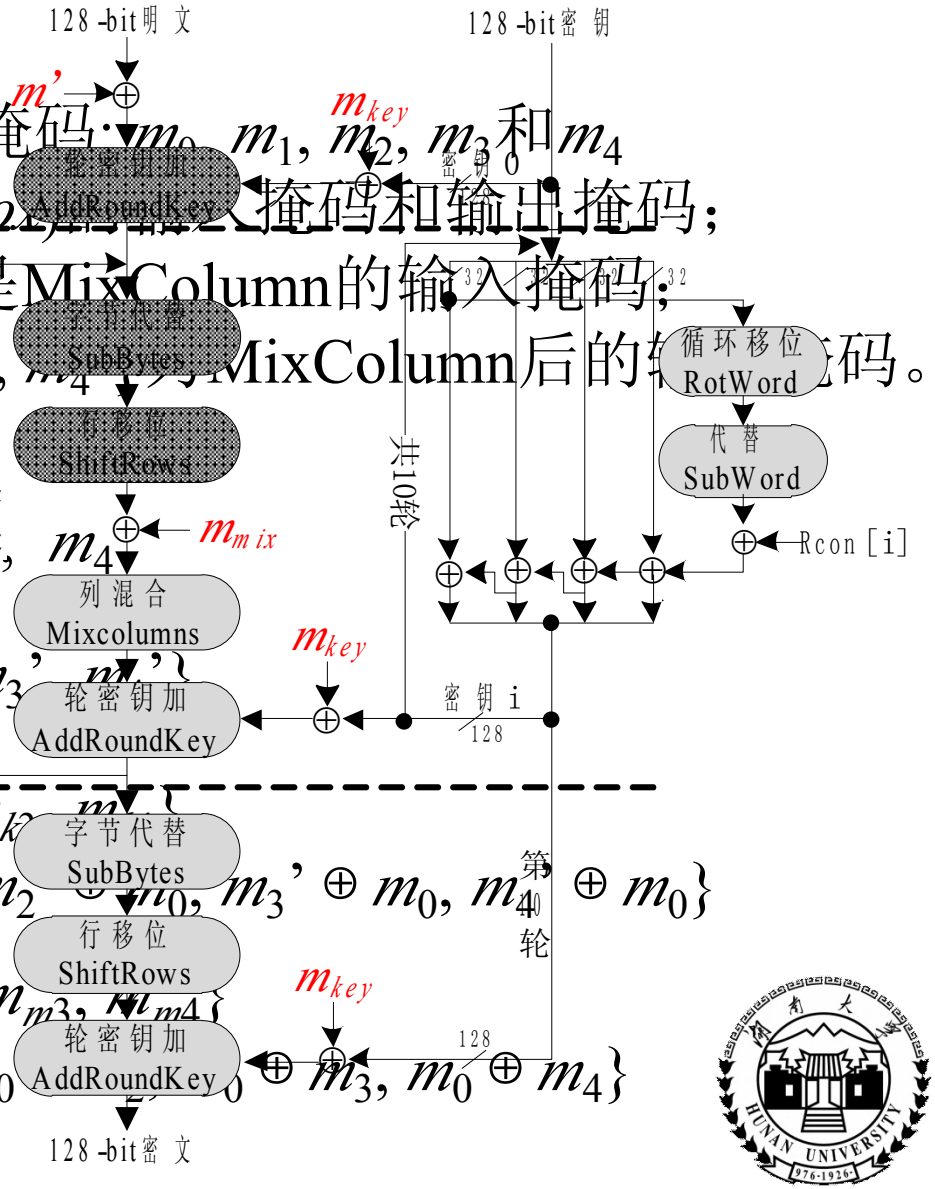
- 采用5个独立8-bit掩码：m_0, m_1, m_2, m_3和m_4
- m_0为新的*S*-盒(*S' box*)的输入掩码和输出掩码；
- m_1, m_2, m_3和m_4则是MixColumn的输入掩码；
- $m' = \{m_1', m_2', m_3', m_4'\}$为MixColumn后的新掩码。

掩码: m_0, m_1, m_2, m_3, m_4

m' $= \{m_1', m_2', m_3', m_4'\}$

m_{key} $= \{m_{k1}, m_{k2}, m_{k3}, m_{k4}\}$
$= \{m_1' \oplus m_0, m_2 \oplus m_0, m_3' \oplus m_0, m_4 \oplus m_0\}$

m_{mix} $= \{m_{m1}, m_{m2}, m_{m3}, m_{m4}\}$
$= \{m_0 \oplus m_1, m_0 \oplus m_2, m_0 \oplus m_3, m_0 \oplus m_4\}$



本文工作 2/2 ——基于秘密共享的AES抗功耗攻击方案

背景

介绍

本文工作

实验&结论

下一步工作

AES *S*-盒

$$S(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \bullet x^{-1} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$\boldsymbol{m}$: $[h, g, f, e, d, c, b, a]$

$\boldsymbol{n}$: $[z, y, x, w, v, u, t, s]$

Linear Function $n=L(m)$:

$s = a + e + f + g + h + 1;$
$t = a + b + f + g + h + 1;$
$u = a + b + c + g + h;$
$v = a + b + c + d + h;$
$w = a + b + c + d + e;$
$x = b + c + d + e + f + 1;$
$y = c + d + e + f + g + 1;$
$z = d + e + f + g + h\,.$

$n_1=\mathcal{L}_1(m_2, p_2, r)$:

$s_1 = a_2 + e_2 + f_2 + g_2 + h_2 + p_2 + r,$
$t_1 = a_2 + b_2 + f_2 + g_2 + h_2 + p_2 + r,$
$u_1 = a_2 + b_2 + c_2 + g_2 + h_2 + r,$
$v_1 = a_2 + b_2 + c_2 + d_2 + h_2 + r,$
$w_1 = a_2 + b_2 + c_2 + d_2 + e_2 + r,$
$x_1 = b_2 + c_2 + d_2 + e_2 + f_2 + p_2 + r,$
$y_1 = c_2 + d_2 + e_2 + f_2 + g_2 + p_2 + r,$
$z_1 = d_2 + e_2 + f_2 + g_2 + h_2 + r\,;$

$n_2=\mathcal{L}_2(m_1, p_1, r)$:

$s_2 = a_1 + e_1 + f_1 + g_1 + h_1 + p_1 + r,$
$t_2 = a_1 + b_1 + f_1 + g_1 + h_1 + p_1 + r,$
$u_2 = a_1 + b_1 + c_1 + g_1 + h_1 + r,$
$v_2 = a_1 + b_1 + c_1 + d_1 + h_1 + r$
$w_2 = a_1 + b_1 + c_1 + d_1 + e_1 + r,$
$x_2 = b_1 + c_1 + d_1 + e_1 + f_1 + p_1 + r,$
$y_2 = c_1 + d_1 + e_1 + f_1 + g_1 + p_1 + r,$
$z_2 = d_1 + e_1 + f_1 + g_1 + h_1 + r\,.$

本文工作 2/2 ——基于秘密共享的AES抗功耗攻击方案

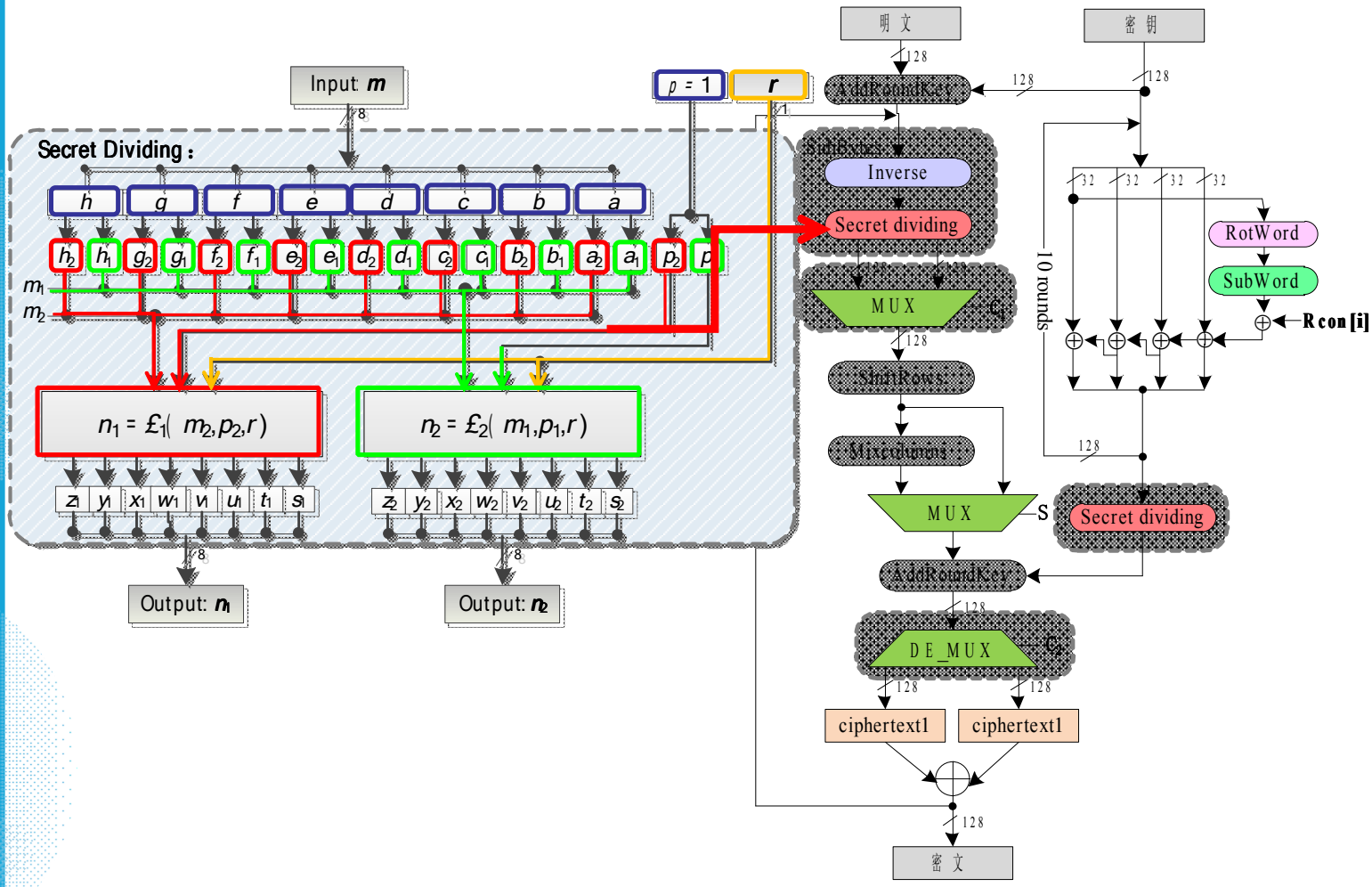
背景

介绍

本文工作

实验&结论

下一步工作



本文工作 2/2 ——基于秘密共享的AES抗功耗攻击方案

背景

介绍

本文工作

实验&结论

下一步工作

正确性：

- **不完整性(**秘密拆分函数至少要独立于输入变量m的一个share**)**：
  - 秘密拆分函数为：$n_1 = \pounds_1(m_2, p_2, r)$、$n_2 = \pounds_2(m_1, p_1, r)$；
- **等值性(**输出share的总和要与期望的输出相等**)**：
  - $\pounds_1$独立于m_1，$\pounds_2$独立于m_2；
  - *SubBytes*变换；
  - *ShiftRows*变换；
  - *Mixcolumns*变换；
  - *AddRoundKey*变换。
- **均衡性(**等概率分布**)**：
  - 当$m = (m_1, m_2)$和$p = (p_1, p_2)$是等概率分布时，经过秘密拆分函数$n_1 = \pounds_1(m_2, p_2, r)$，$n_2 = \pounds_2(m_1, p_1, r)$运算后的输出结果$n_1$，$n_2$也满足等概率分布。

本文工作 2/2 ——基于秘密共享的AES抗功耗攻击方案

背景

介绍

本文工作

实验&结论

下一步工作

安全性：

- **抗HO-DPA攻击：**
  - 输入被分成了2个share；
  - $\mathcal{L}1$独立于m_1、$\mathcal{L}_2$独立于m_2；
  - HO-DPA攻击仅能得到两组独立无关的中间值，也就不能通过对这两组中间值的统计分析来获取信息。

- **抗glitch攻击：**
  - 由于输入被秘密拆分函数分成了两个share，并且分别进行计算，因此发生在一个share上的glitch与发生在另一个share上的glitch并不同步；
  - 引入了随机值r；
  - 难以通过一组share上的glitch攻击来获取整个电路的信息。

实验&结论 1/4 ——实验方案

背景

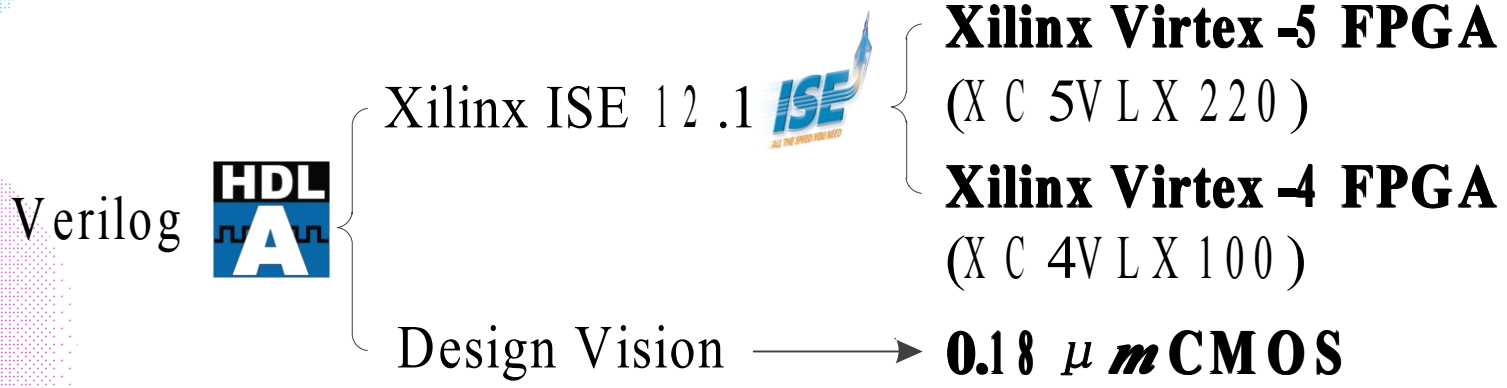
介绍

本文工作

实验&结论

下一步工作

本章采用硬件描述语言Verilog实现了算法的逻辑设计，并用分别使用Xilinx ISE 12.1将设计实现在Xilinx Virtex-5 FPGA (XC5VLX220)和Xilinx Virtex-4 FPGA (XC4VLX100)上，以及使用Design Vision将设计实现在0.18μm CMOS上。



实验&结论 1/4 ——实验方案

背景

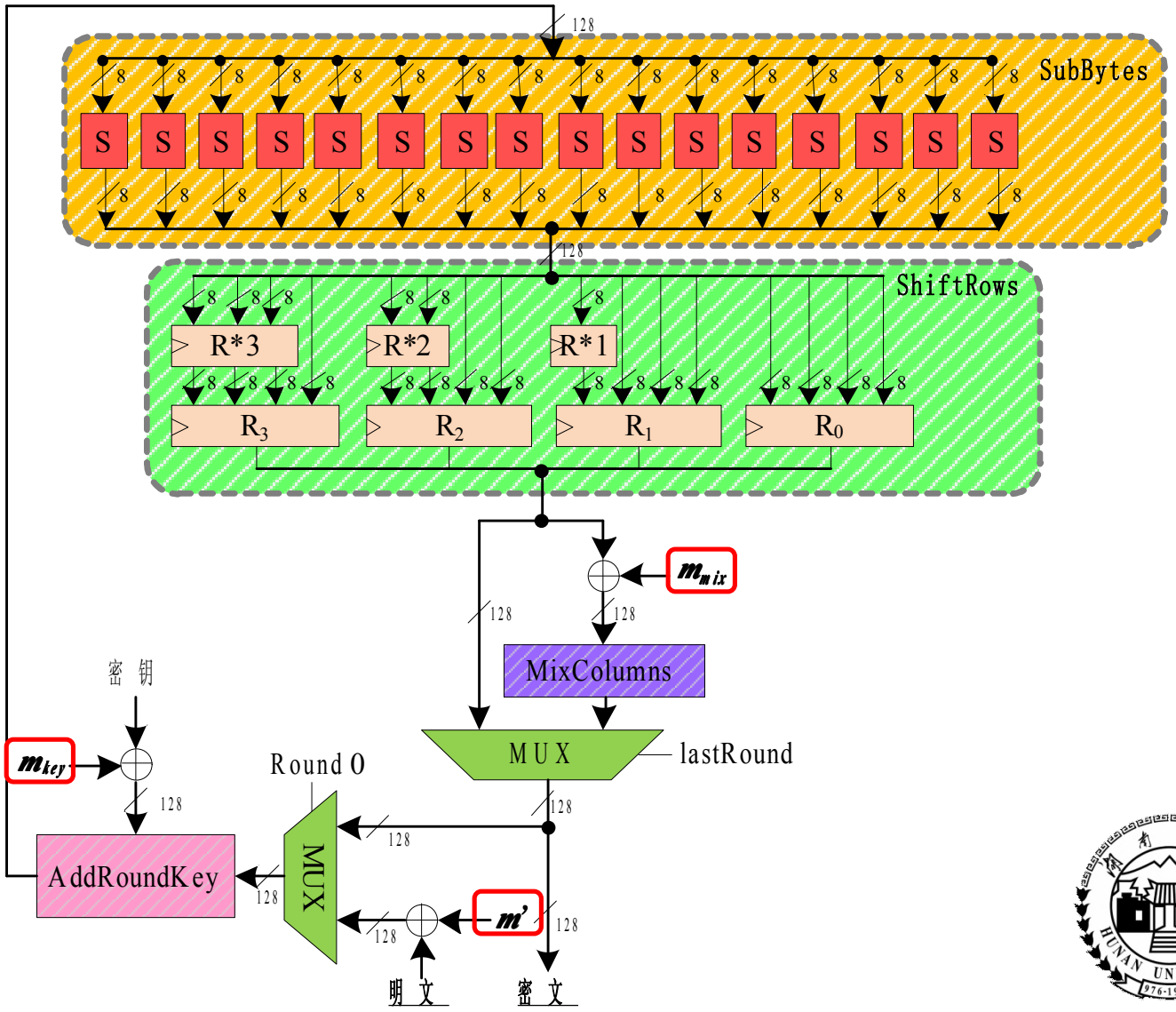
介绍

本文工作

实验&结论

下一步工作

图1 基于掩码的AES抗功耗攻击方案数据通路图



实验&结论 1/4 ——实验方案

背景

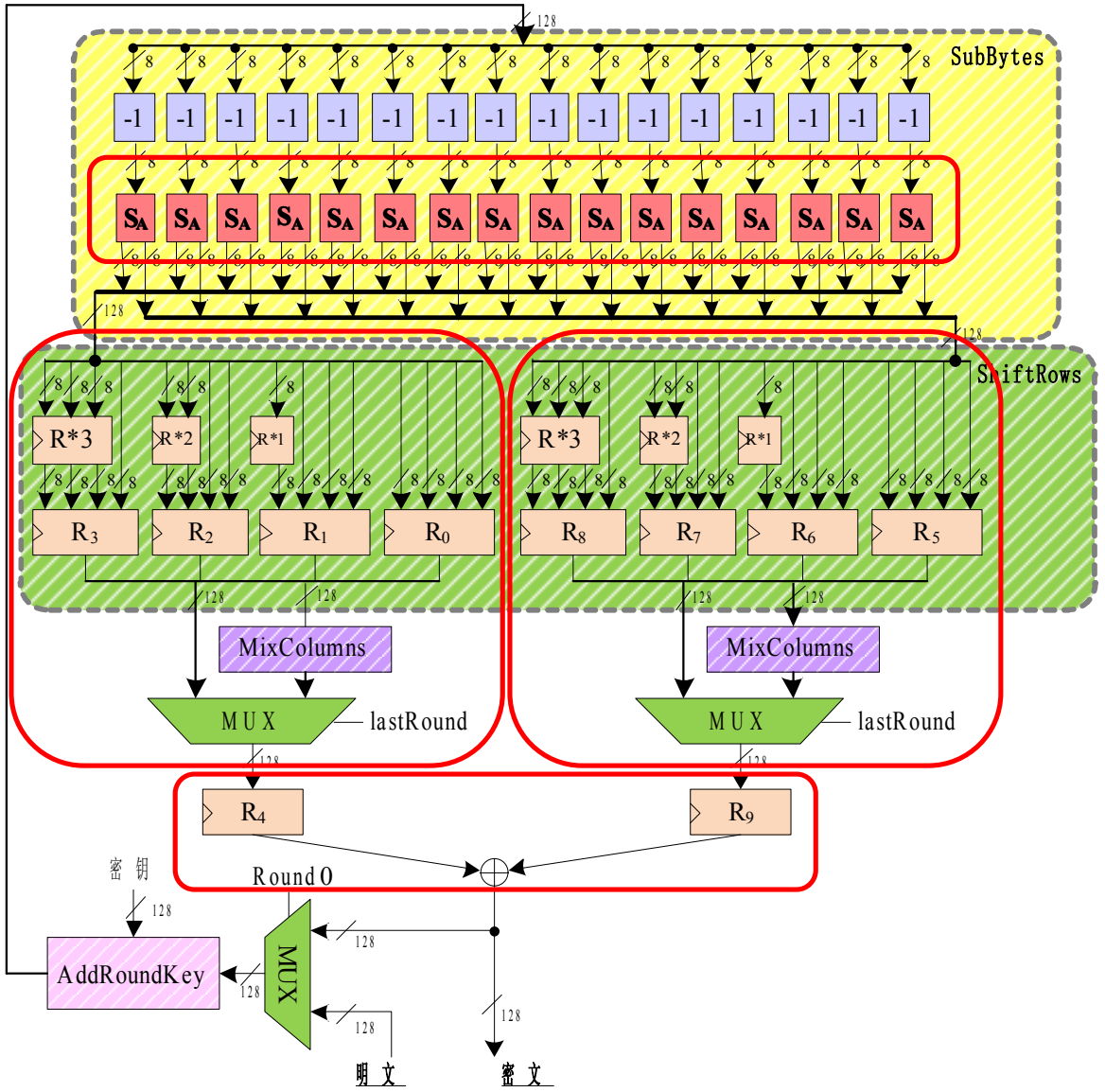
介绍

本文工作

实验&结论

下一步工作

图2 基于秘密共享的AES抗攻耗攻击方案数据通路图(并行方案)



实验&结论 1/4 ——实验方案

背景

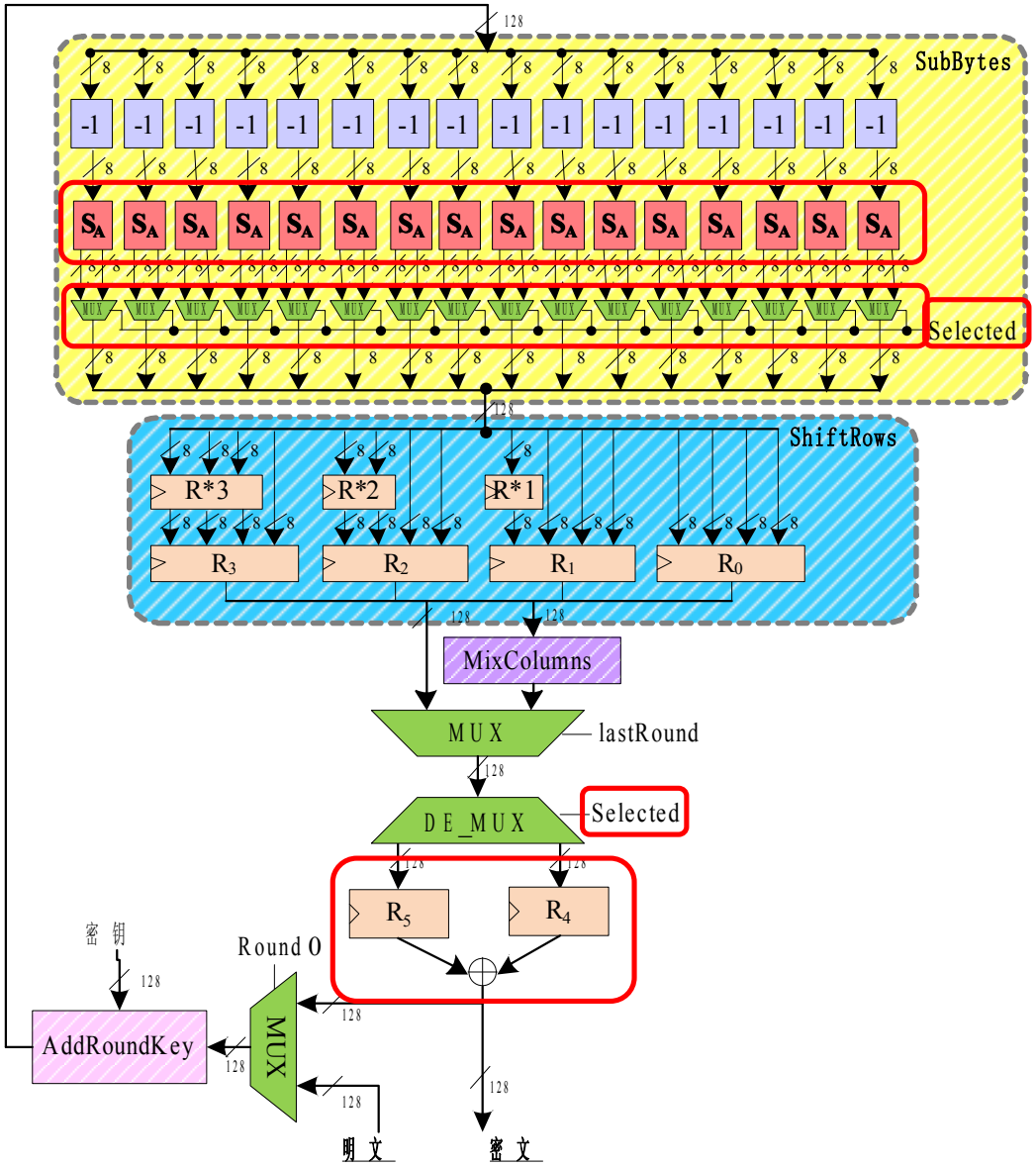
介绍

本文工作

实验&结论

下一步工作

图3 基于秘密共享的AES抗攻耗攻击方案数据通路图(时钟控制方案)



实验&结论 1/4 ——实验结果

表1. *S*-盒在FPGA平台的执行结果比较

	平台	方法	面积 (slices)	延迟 (ns)
Kamoun[45]	Xilinx Virtex-4	算法级掩码	100	16.670
	Xilinx Virtex-4	秘密共享	70	9.110
本文设计	Xilinx Virtex-5	掩码	127	14.299
	Xilinx Virtex-5	秘密共享	30.00% lesser	45.35% lesser

表2. *S*-盒在ASIC平台的执行结果比较

	平台	方法	面积 (gates)	延迟 (ns)
Baek[25]	0.18μm CMOS	算法级掩码	1,023	27.0
	0.18μm CMOS	秘密共享	609	10.0
本文设计			40.47% lesser	62.96% lesser

实验&结论 1/4 ——实验结果

背景

介绍

本文工作

实验&结论

下一步工作

表 3 AES在FPGA平台的执行结果比较

	平台	方法	面积 (slices)	速度 (MHz)	吞吐量 (Mbps)
Matsumoto[52]	Xilinx Virtex-2	门级掩码	2,744	25.4	155
Popp[49]	Xilinx Virtex-2	门级掩码	12,691	15.7	96
Trichina[36]	Xilinx Virtex-2	与门掩码	3,017	44.0	512
Nikova[39]	Xilinx Virtex-2	秘密共享	10,619	63.7	337
Mentens[53]	Xilinx Virtex-2	混合掩码	4,452*	23.0	29
Akkar[24]	Xilinx Virtex-E	乘法掩码	4,175	43.8*	140
Oswald[54]	Xilinx Virtex-E	加法掩码	3,580	43.9*	157
Kamoun[45]	Xilinx Virtex-4	算法掩码	2,281	137.0	
本文设计	Xilinx Virtex-4	秘密共享P	2,618	197.2	2294
	Xilinx Virtex-4	秘密共享P	2,618	197.2	2294
	Xilinx Virtex-5	掩码	4,992	116.0	1350
	Xilinx Virtex-5	秘密共享C	795	272.9	1588
	Xilinx Virtex-5	秘密共享P			

3.28 % lesser

1.25 tim faster

1.6 times faster

实验&结论 1/4 ——实验结果

背景

介绍

本文工作

实验&结论

下一步工作

表4 AES在ASIC平台的执行结果比较

	平台	方法	面积 (gates)	速度 (MHz)	吞吐量 (Mbps)
Lin[46]	0.18μm CMOS	门级掩码	20,100	35.6*	111.0
Trichina[47]	0.18μm CMOS	与门掩码	18,600	46.4*	145*
Tiri[48]	0.18μm CMOS	门级掩码	30,300	13.0*	40.6*
Popp[49]	0.18μm CMOS	门级掩码	45,850	25.0*	78.0*
Back[25]	0.18μm CMOS	算法级掩码	25,700	17.5*	14.0*
Xinjian[50]	0.18μm CMOS	布尔掩码	49,000	100	900
Tiri[51]	0.18μm CMOS	门级掩码	245,000	85	-
本文设计	0.18μm CMOS	秘密共享P	17,117	143	1664.0*
	0.18μm CMOS	秘密共享P	17,117	143	1664.0*

7.79 % lesser

1.43 times faster

1.85 times larger

实验&结论 1/4 ——实验结论

背景

介绍

本文工作

实验&结论

下一步工作

- ***GF*(2⁴)域上基于掩码的AES抗功耗攻击方案**
  - 面积居中；
  - 吞吐量比现有设计的大；
  - 速度也具有一定的竞争力。
- **基于秘密共享的AES抗攻耗攻击方案**
  - 面积比现有设计小；
  - 速度比现有设计快；
  - 吞吐量比现有设计大；
  - 拆分份额比现有设计少。

	面积	速度	吞吐量
时钟控制方式	√	√	
并行方式			√

在实际应用中，用户可根据自己的实际需要进行选择。

下一步工作

背景

介绍

本文工作

实验&结论

下一步工作

- **基于秘密共享的AES加密算法的深度研究**
  (考虑对AES的求逆变换进行秘密拆分，通过引入高斯变换等函数对拆分进行随机化，从而进一步加大攻击难度)
- **采用多种防御技术相结合的的AES加密算法研究**
  (考虑采用隐藏、乱序、掩码、秘密共享策略中的两种或多种防御算法相结合的方式对AES进行加密)
- **采用实际的功耗攻击分析**
  (考虑使用功耗攻击专用板SASEBO进行功耗轨迹的抓取和分析，从而在理论和实践两方面对算法的安全性进行分析)

HUNAN UNIVERSITY

湖南大学

谢谢答辩组各位老师！