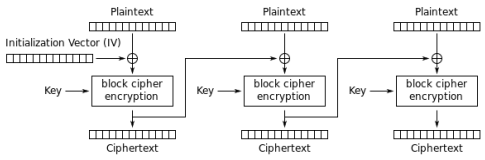
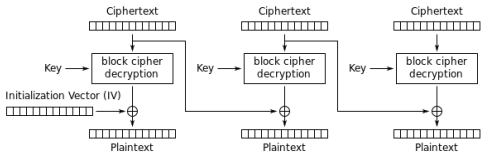
Pride

CBC

- CBC mode 计算规则
  1) 包括密钥编排、加密、解密三部分；
  2) RAM：明文、主密钥、轮子密钥、初始向量、解密辅助数据（保存临时密文），密钥编排辅助数据；
  3) Flash(data)
     a) 数据：不包括明文、主密钥、初始向量 vector，直接对 RAM 中的数据进行初始化；包括轮常量（比如 Simon 中的 Z、Rectangle 中密钥编排的常量）；
  4) Flash(code) & Time
     a) 密钥编排
        - 起始状态：主密钥初始化完成，轮常量初始化完成；
        - 结束状态：主密钥中数据不变，轮子密钥全部计算完成并保持在 RAM 中；
        - 包括内容：~~主密钥复制到轮密钥前 16bytes（以 Simon64/128 为例）~~、轮函数计算，程序框架（寄存器初始化，循环控制，子程序调用与返回，辅助数据更新）；
     b) 加密
        - 起始状态：明文、初始向量初始化完成，轮子密钥计算完成；
        - 结束状态：所有明文被密文覆盖、RAM 中初始向量和轮子密钥保存不变；
        - 包括内容：加载明文、加载初始向量、异或向量、轮函数加密、写回密文同时更新寄存器中向量、程序框架（寄存器初始化，不同 Block 间循环控制、同一 Block 内轮数循环控制，子程序调用与返回）；
     c) 解密
        - 起始状态：明文被密文覆盖，初始向量、轮子密钥没被修改；
        - 结束状态：密文被恢复成明文、RAM 中初始向量和轮子密钥保存不变；
        - 包括内容：加载密文、加载初始向量、密文复制（解密过程中密文会被覆盖，而在下一个 Block 解密时需要用到本次的密文，因此需要在密文被明文覆盖前先复制）、轮函数解密、异或向量、写回明文、使用复制的密文更新寄存器中向量、程序框架（寄存器初始化，不同 Block 间循环控制、同一 Block 内轮数循环控制，子程序调用与返回）；



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

			Pride
RAM	总计		244 bytes
	向量		8
	明文/密文		128
	主密钥		16
	轮子密钥		84
	保存密文		8
Flash(data)	总计		0 bytes
	轮常量		0
Flash(code) /Time、	三部分总计 Flash(code)/Time		802 bytes 54151 cycles
	密钥编排	总计	102/593
		装载主密钥	44/90
		常量初始化	8/4
		加载密钥固定部分	8/8
		乘法	8/160(8*20)
		加法	8/80(4*20)
		存储主密钥	8/160(8*20)
		程序框架	18/91
	加密	加密总计	330/25915
		加载向量	16 / 16
		向量异或	16 / 128(8*16)
		加载明文	16 / 256(16*16)
		加载白化密钥	32 / 512(32*16)
		异或白化密钥	32 / 256(16*16)
		S 盒	40 / 6400(20*20*16)
		L 层	72 / 10944(36*19*16)
		轮密钥异或	16 / 2560(8*20*16)
		加载轮密钥固定部分	8 / 128(8*16)
		加载轮密钥变化部分	8 / 2560(8*20*16)
		更新向量	8 / 64(4*16)
		写回密文	16 / 256(16*16)
		程序框架	50 / 1835
	解密	解密总计	370/27643
		加载向量	16 / 16
		向量异或	16 / 128(8*16)
		加载密文	16 / 256(16*16)
		临时保存密文	16 / 256(16*16)
		加载白化密钥	32 / 512(32*16)

批注 [a1]: 固定部分用 4bytes 存储；不固定的用 80bytes 存储

批注 [a2]: 22 条指令
44 bytes = 22*2
90 cycles = 7 + (8*4+3*2+1) + 5 + (8*4+3*2+1)

批注 [a3]: 9 条指令
18 bytes = 9*2
91cycles=2+2+1+20+20+(19*2+1)+3+4
2 条获取密钥地址指令，执行 1 次；
1 条 sbiw 指令，执行 1 次；
1 条轮数寄存器初始化指令，执行 1 次；
1 条计数器加 1 指令，执行 20 次；
1 条比较指令，执行 20 次；
1 条 brne 指令，19 次执行跳转，另外 1 次不跳转；
1 条调用指令，执行 1 次；
1 条返回指令，执行 1 次

批注 [a4]: 解密和加密的不同在于以下两点：
1.解密前需要临时保存密文用于下一轮的解密。以下均为 1 个 block 比较：
相比加密多了临时保存密文的过程，需要 8 条指令、16 cycles；同时解密结束后需要 8 条指令（ld）重新加载密文，这比加密的 4 条指令（movw）多了 12 cycles；同时，保存密文和重新加载密文需要多 4 条获取地址指令、4 cycles；
2.逆 L 层比加密多了 4 条指令；

批注 [a5]: 绿色底纹标出的地方表示解密和加密的不同之处。

		异或白化密钥	32 / 256(16*16)
		逆 S 盒	40 / 6400(20*20*16)
		逆 L 层	80 / 12160(40*19*16)
		轮密钥异或	16 / 2560(8*20*16)
		加载轮密钥固定部分	8 / 128(8*16)
		加载轮密钥变化部分	8 / 2560(8*20*16)
		更新向量	16 / 256(16*16)
		写回明文	16 / 256(16*16)
		程序框架	58 / 1899

[1] Pride 加密程序框架

50 bytes

1835 cycles

- 2 条获取向量地址指令，执行 1 次； (4/2)
- 1 条加密块数初始化指令，执行 1 次； (2/1)
- 2 条获取明文地址指令，执行 1 次； (4/2)
- 4 条获取白化密钥地址，执行 16 次； (8/64)
- 1 条轮数初始化指令，执行 16 次； (2/16)
- 2 条获取固定密钥地址，执行 16 次； (4/32)
- 2 条获取变化密钥地址，执行 16 次； (4/32)
- 1 条轮数比较指令，执行 20*16 次； (2/320)
- 1 条轮数内 breq 指令，跳转 16 次，不跳转 19*16 次； (2/336)
- 1 条轮数加 1 指令，执行 19*16 次； (2/304)
- 1 条 rjmp 指令，执行 19*16 次； (2/608)
- 1 条明文地址更新指令，执行 16 次； (2/32)
- 1 条块数加 1 指令， 执行 16 次； (2/16)
- 1 条块间比较指令，执行 16 次； (2/16)
- 1 条块间 breq 指令，跳转 1 次，15 次不调整； (2/17)
- 1 条 rjmp 指令，执行 15 次； (2/30)
- 1 条子程序调用指令，执行 1 次； (2/3)
- 1 条返回指令，执行 1 次； (2/4)

[2] Pride 解密程序框架

58 bytes

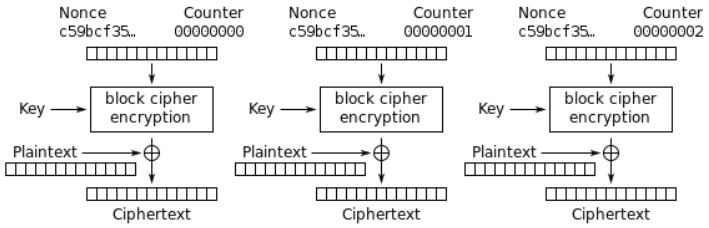
1899 cycles

- 2 条获取向量地址指令，执行 1 次； (4/2)
- 1 条加密块数初始化指令，执行 1 次； (2/1)
- 2 条获取明文地址指令，执行 1 次； (4/2)
- 4 条获取临时密文地址，执行 16 次； (8/64)
- 4 条获取白化密钥地址，执行 16 次； (8/64)
- 1 条轮数初始化指令，执行 16 次； (2/16)
- 2 条获取固定密钥地址，执行 16 次； (4/32)
- 2 条获取变化密钥地址，执行 16 次； (4/32)
- 1 条轮数比较指令，执行 20*16 次； (2/320)

- 1 条轮数内 breq 指令，跳转 16 次，不跳转 19*16 次； (2/336)
- 1 条轮数加 1 指令，执行 19*16 次； (2/304)
- 1 条 rjmp 指令，执行 19*16 次； (2/608)
- 1 条明文地址更新指令，执行 16 次； (2/32)
- 1 条块数加 1 指令， 执行 16 次； (2/16)
- 1 条块间比较指令，执行 16 次； (2/16)
- 1 条块间 breq 指令，跳转 1 次，15 次不调整； (2/17)
- 1 条 rjmp 指令，执行 15 次； (2/30)
- 1 条子程序调用指令，执行 1 次； (2/3)
- 1 条返回指令，执行 1 次； (2/4)

CTR：Low Flash 的实现（采用循环加密两个 Block）

- CTR mode 计算规则
  1) 只包括加密，没有密钥编排、解密部分；
  2) 没有 nonce，直接对计数器加密后与明文异或；
  3) RAM：明文，计数器；
  4) Flash(data)
     a) 数据：只包括轮子密钥，明文和计数器直接在 RAM 中进行初始化；
  5) Flash(code) & Time：加密（具体包括内容见下表）的代码和时间；



Counter (CTR) mode encryption

			Pride
RAM	总计		24 bytes
	明文/密文		16
	计数器		8
Flash(data)	总计		92 bytes
	轮常量		92
Flash (code) Time	总计 Flash(code)/Time		340 bytes 3468 cycles
	加密	加载计数器	16 / 16
		复制计数器	8 / 4
		计数器加 1	2 / 1
		加载白化密钥	32 / 96(48*2)
		异或白化密钥	32 / 32(16*2)
		加载轮密钥固定部分	8 / 24(12*2)
		加载轮密钥变化部分	8 / 480(12*20*2)
		S 盒	40 / 800(20*20*2)
		L 层	72 / 1368(36*19*2)
		轮密钥异或	16 / 320(8*20*2)
		加载明文	16 / 32(16*2)
		明文异或计数器	16 / 16(8*2)
		重新加载计数器	8 / 8(4*2)
		写回密文	16 / 32(16*2)
		程序框架	50 / 239

批注 [a6]:
8 bytes 白化密钥：轮子密钥中有一半是不变的，因此只用 4 bytes 保存不变的部分；剩下变化的部分用 80 bytes(4 bytes * 20)保存

[3] Pride CTR 加密程序框架

50 bytes

239 cycles

- 2 条指令获取计数器地址，执行 1 次； (4/2)
- 2 条指令获取明文地址，执行 1 次； (4/2)
- 1 条指令初始化块数，执行 1 次； (2/1)
- 4 条指令获取白化密钥地址，执行 2 次； (8/8)
- 1 条指令初始化轮数，执行 2 次； (2/2)
- 2 条指令获取固定轮子密钥地址，执行 2 次； (4/4)
- 2 条指令获取变化轮子密钥地址，执行 2 次； (4/4)
- 1 条比较指令，执行 20*2 次； (2/40)
- 1 条 brne 指令跳转 2 次，不跳转 19*2 次； (2/42)
- 1 条轮数寄存器加 1 指令，执行 19*2 次； (2/38)
- 1 条轮数内 rjmp 指令，执行 19*2 次； (2/76)
- 1 条 adiw 指令，执行 2 次； (2/4)
- 1 条块数加 1 指令，执行 2 次； (2/2)
- 1 条块间比较指令，执行 2 次； (2/2)
- 1 条块间 breq 指令，跳转 1 次，1 次不调整； (2/3)
- 1 条块间 rjmp 指令，执行 1 次； (2/2)
- 1 条子程序调用指令，执行 1 次； (2/3)
- 1 条返回指令，执行 1 次； (2/4)