基于三种常见算法的RKE/PKE系统

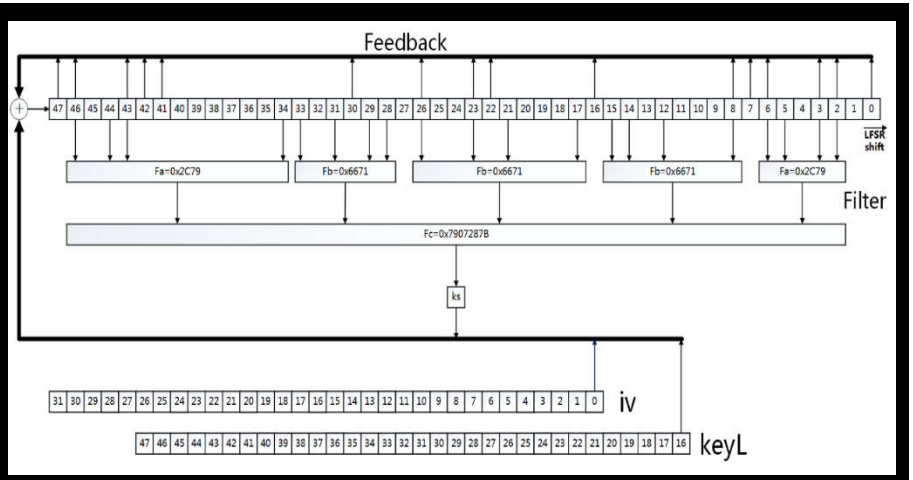
--Hitag2,Hitag3,Megamos

RKE系统的数据流频率一般为2kHz-20kHz，数据长度为64到128位，其中包括前导码、按键信息、钥匙ID、滚码、状态信息、校验码等部分。

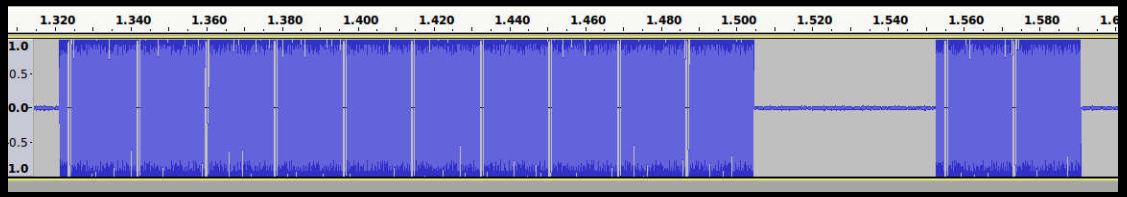
PKE系统中钥匙端发送的是高频信号，通常采用315MHz和434MHz频段，汽车端发送的是低频信号，一般采用125kHz。

典型的PKE系统如右图所示。

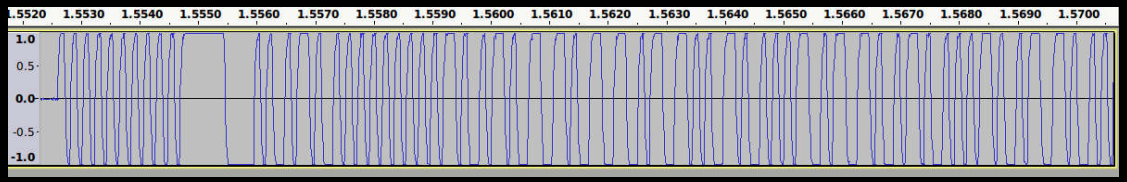
Hitag2



Hitag2算法是NXP公司提出的一种序列密码，其密钥长度较短，为48位。参与运算的还有32位的身份识别ID、32位的计数器ctr和8位按键btn。iv一般由ctr||btn得到。



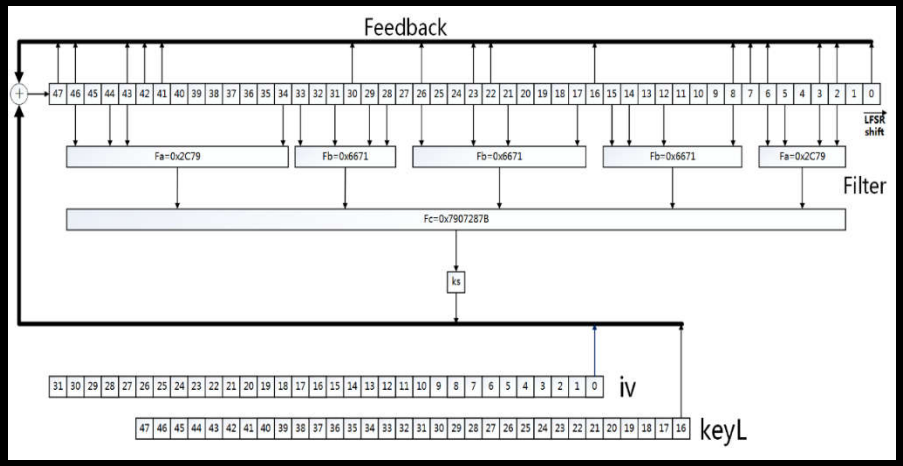
为明确已知的数据量和数据格式，截获了钥匙与车之间通讯的信号如上图所示。下图是取其中一个重复块放大，电压1.0V代表1，-1.0V代表0。



数据格式：

0xcc	btn	UID	ks	lctr	chk

A块与B块的数据都满足以上格式。A块与B块之所以要重复多次，是为了防止信息错误与丢失。A块中的btn是实际的钥匙按键信息，而B块相对于A块的唯一区别就在于btn为0x00。我们经过实际验证发现，A块为有效信息，B块为结束标志，不承担传输数据的功能，但是会使得计数器+1。

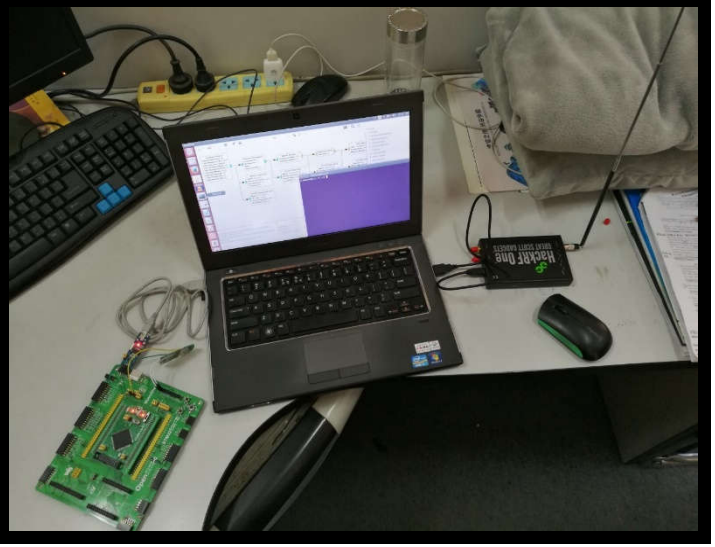


针对这一密码算法，我们将重点放在了它的中间变量state上，利用短时间内监听的多条数据的iv高位相同的特征，来判断不同state生成密文的概率，最后高概率的state个数在我们能接受的范围内。

通过HackRF监听信号并发射我们自己定义的信号

电脑通过串口以115200bps的码率给STM32传送数据

主控是STM32F407开发板，STM32控制射频发射模块

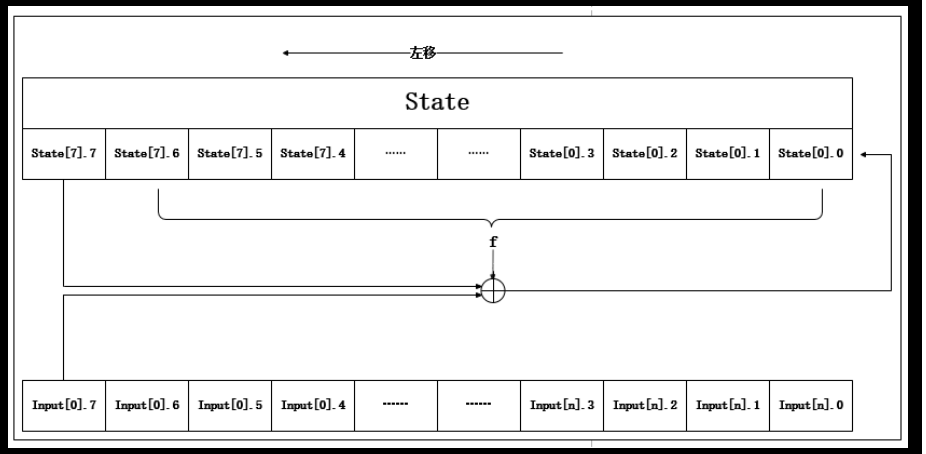



车门是关着的，拉不开

我们破解了宝马X1 2014款和广汽传祺GS4两款的RKE系统和防盗控制系统。这意味着我们不仅可以打开车门，还可以将车开走而不引人注意。

Hitag2算法在国内除了上述两种车型，还有非常多厂商的众多车型在使用，包括雪佛兰科鲁兹、奇瑞瑞虎、长城炫丽、吉利远景、大众途锐、华泰B11车、现代圣达菲、雪铁龙、标致等知名品牌。

Hitag3

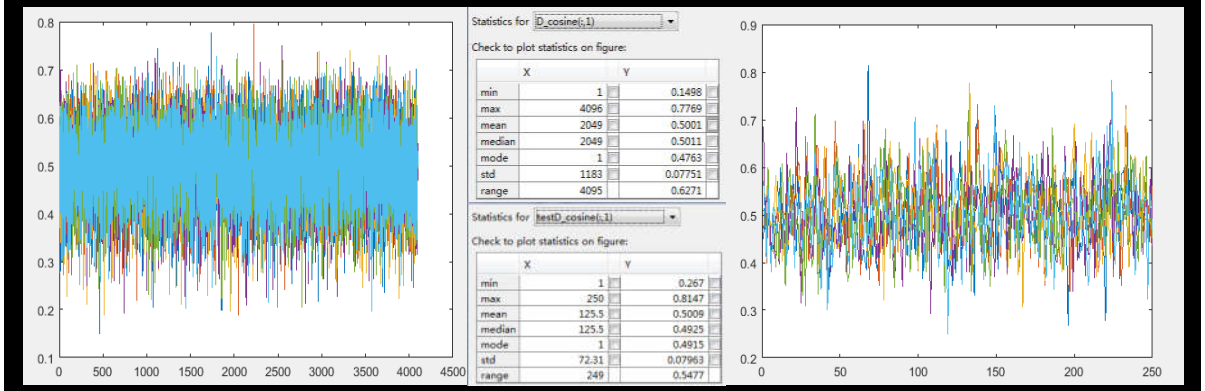


Hitag3算法是一种序列密码，其96位密钥长度保证了算法的安全性，目前为止，尚未有公开的相关破解。参与加密的除了96位密钥外，还有64位challenge、32位IDE。国内可见的主要有夏利N5、N7、威志V5、现代途胜、铃木启悦、本田冠道等。

关键函数f代表的是一个查找表的输出，这导致了一个多对一的映射，使得f的输出输入对之间可能引入不均匀分布，又因为数据量巨大，我们无法得出它的统计规律。

Hitag3算法最终得出的密文分为两部分，分别是被称为mac部分的16位鉴权信息和被称为data部分的48位认证数据。为了提高安全性，这两部分的输出被分开了。在它们的输出之间，密钥Key的高64位被重新滚入加密。

为了减少一些计算量，我们试图找出一些规律，因此尝试了包括杰拉德距离、余弦距离、欧式距离、相关系数等在内的许多距离模型。下面是余弦距离的测试结果。

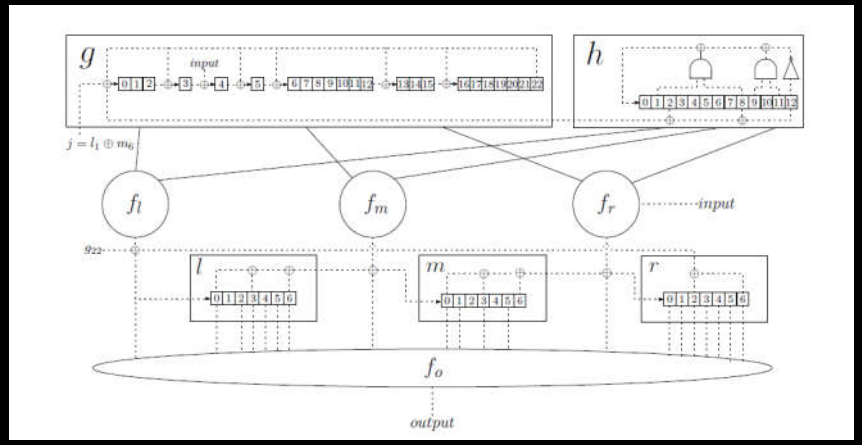


NIST随机性测试结果

测试针对的是8000个state拼接的512000位的数据

针对每个state做了部分测试，基本都符合随机性。

Megamos



Megamos密码算法的密钥有96位，它的state被分割为g、 h、 l、 m、 r 5个部分。除了Key以外，本算法的加密还需要一个激励，我们称之为nc，nc共有56位，我们可以通过窃听钥匙和immobilizer之间的通信得到。

我们记中间状态state：s=<g，h，l，m，r>

预初始化：

$p=nc_0 \ldots nc_{55}+k_{40} \ldots k_{95} \mod 256$

$q=(p_2 \ldots p_{45}) \oplus (p_8 \ldots p_{51}) \oplus (p_{12} \ldots p_{55})$

$t=init(q_{20} \ldots q_{42}, q_0 \ldots q_{19})$

初始化：

$g = t_0 \ldots t_{22}$

$h = 0p_0 \ldots p_{11}$

$l = t_{23} \ldots t_{29}$

$m = t_{30} \ldots t_{36}$

$r = t_{37} \ldots t_{42} q_{43}$

g有23位，h有13位，l、m、r分别有7位

对于下一状态s’=<g’，h’，l’，m’，r’>，有：

$g' = G(g, i, l_1 \oplus m_6 \oplus h_2 \oplus h_8 \oplus h_{12})$

$h' = H(h)$

$l' = a l_0 \ldots l_5$

$m' = b m_0 \ldots m_5$

$r' = c r_0 \ldots r_5$

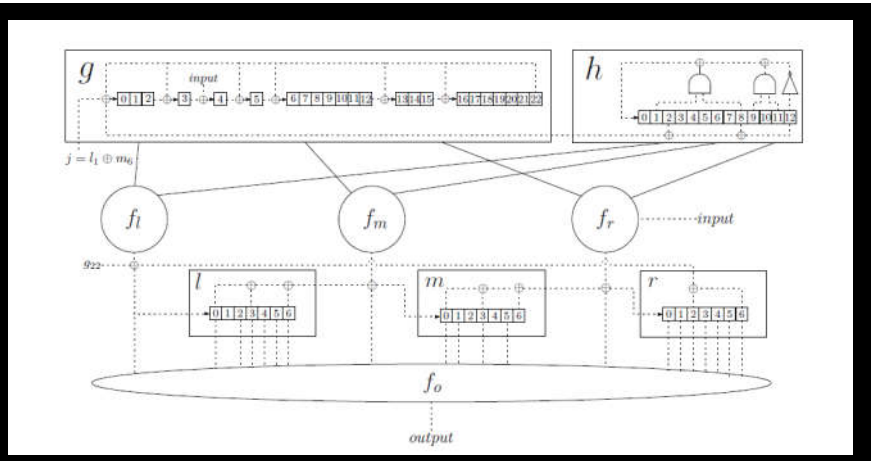
其中：

$a = f_l(g_0 g_4 g_6 g_{13} g_{18} h_3) \oplus g_{22} \oplus r_2 \oplus r_6$

$b = f_m(g_1 g_5 g_{10} g_{15} h_0 h_7) \oplus l_0 \oplus l_3 \oplus l_6$

$c = f_r(g_2(g_3 \oplus i) g_9 g_{14} g_{16} h_1) \oplus m_0 \oplus m_3 \oplus m_6$

主旨：分为g、h和l、m、r两部分



对g、h部分：
遍历h，对每个h生成一个表Th，共有2^{12}个Th表，在选好索引的情况下，每个Th表需要256M存储空间
对l、m、r部分：
遍历lmr，通过15位0输入来验证，有计算复杂度2^{37}，对所有h共有复杂度2^{12+37}

目前，我们在普通的PC上已经针对其中一个h值验证了破解的可能性，利用现有的超算，可以在十分钟内完成对Megamos算法密钥的提取。

诸如大众、奥迪、中华、本田、别克、沃尔沃等厂商的绝大部分车型的防盗控制系统都在使用这一加密算法。

```
#include "encrypt.h"

u32  *p0=(u32 *)calloc((1<<24),sizeof(u32));
struct gtable{u32 data;u32 next;};
struct gtable *current=(struct gtable *)calloc((1<<23),sizeof(u64));

u32  temp_h = 0x0d41;
u32 _h = temp_h;

u8 nolmr_f0(u8 a,u8 b,u8 c,u8 _l,u8 _m,u8 _r){ ... }

//对一个给定的j，f最多重复有150个
void generate(u8 j){ ... }

void tradition(u32 g,u8 _l,u8 _m,u8 _r){ ... }

void generate_lmr(u8 real_j){ ... }

int main()
{
    u32 g = cipher_init();
    u8 j=round(g);

    generate(j);
    generate_lmr(j);

    free(p0);
    free(current);
    return 0;
}
```

1、Hitag2加密算法中短时间内的iv相近导致了漏洞

2、Hitag3加密算法的随机性经检验是没有问题的，目前还是比较安全的

3、Megamos加密算法将state分为5个部分，整体上增加了安全性，但对特定的方法是一个比较大的弱点

1、在不改变硬件环境的情况下，只从软件上着手，减少钥匙与车之间信息交互的数量

2、推荐使用AES加密算法，这是大势所趋。AES加密算法密钥长度支持128、192、256位三种。目前，对128位长度密钥的代数差分攻击也只能降低8位的计算复杂度。

Thank You!