IoT设备中的分组密码研究

——以Chaskey、 Simon和Speck为例

|高亨利 1901210713|刘存展 1901210442|胡兆杰 1901210403|

目录
CONTENTS

01

PART 01

第一部分

研究背景

背景

思科IBSG的最新白皮书估计，到2020年，将有500亿个设备连接到互联网，这意味着在不远的将来，每个人都将被数十种传感器设备所包围，从互联网到物联网的这种演进将对我们的日常生活产生巨大影响，并改变我们与周围的物理世界的交互方式。 然而，显而易见的是，连接到Internet的500亿智能设备给其所有者或用户的安全和隐私带来了前所未有的挑战。众所周知，对称密码系统在物联网的安全领域中扮演着重要角色， 但是物联网硬件设备的计算和存储资源十分有限，这就需要对密码体系进行更加细致有效的设计，保证在物联网设备中有限的资源上能够稳定运行。

在本篇报告中，我们选取了3种在IoT设备中性能表现优异并且安全性较高的密码方案，即Chaskey、Simon和Speck，来进行简单的分析和介绍。

02

PART 02

第二部分

密码算法介绍

01 Chaskey　　02 Simon & Speck

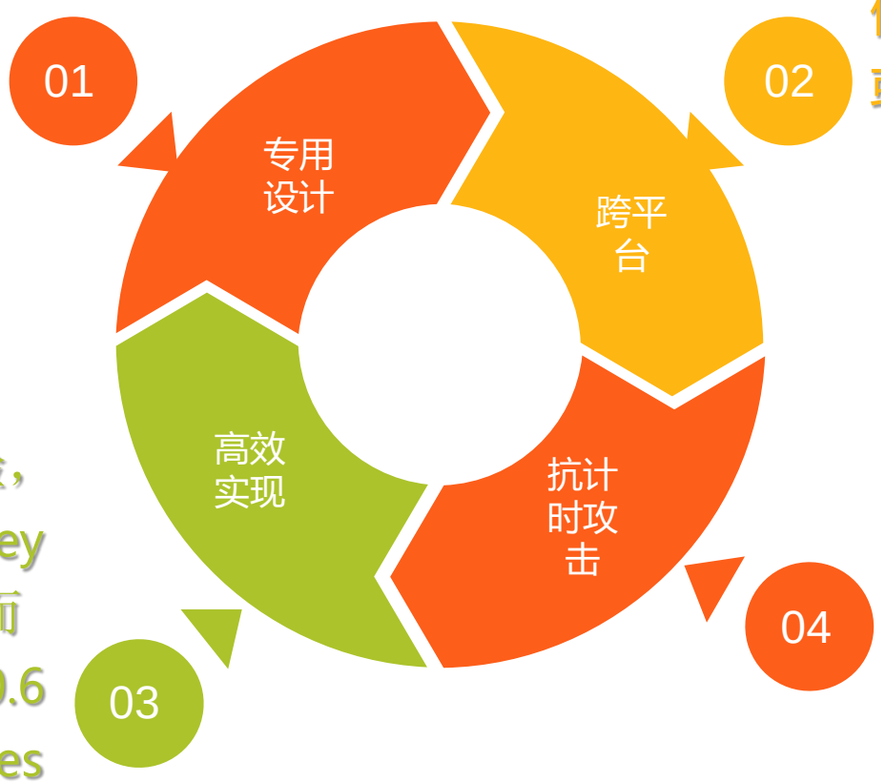
密码算法介绍——Chaskey

- Chaskey主要用于克服现有密码算法在微控制器上难以实现的问题。它是基于排列的MAC算法
- Chaskey算法中使用的密钥K长度为128比特
- Chaskey算法将消息M分成大小相等的128比特的块进行置换π
- Chaskey 的 置 换 采 用 Addition-Rotation-XOR设计方式

Chaskey的功能特点

Chaskey是用于32位微控制器体系结构的专用设计。加法和异或操作都是对于32bit的字为单位执行的，并且它们在32位的体系结构下都只需要通过一条指令就可以实现。

我们考虑到某些微控制器没有支持可变长度的循环移位和位平移。所以我们通过选择一些8的倍数作为移位位数，使得这些操作可以通过寄存器上的8位或16位交换（swap）操作有效实现。



算法运行在ARM Cortex-M4上试验，对于长消息（≥128字节），Chaskey算法运行仅需要7.0 cycles/byte，而对于短消息（16字节）也仅需要10.6 cycles/byte，并且只占用了402bytes的ROM。

在所有的微控制器架构上，Chaskey的每条指令的执行时间都是固定的。且执行的时钟周期数仅取决于消息的长度。因此，Chaskey具有固有的抵御计时攻击的安全性。

Chaskey算法流程

Chaskey使用n比特长度的密钥K为任意大小的消息M生成一个标签τ，并且标签τ的长度t小于等于n。我们通过算法1和2为密钥K生成两个子密钥K1和K2。

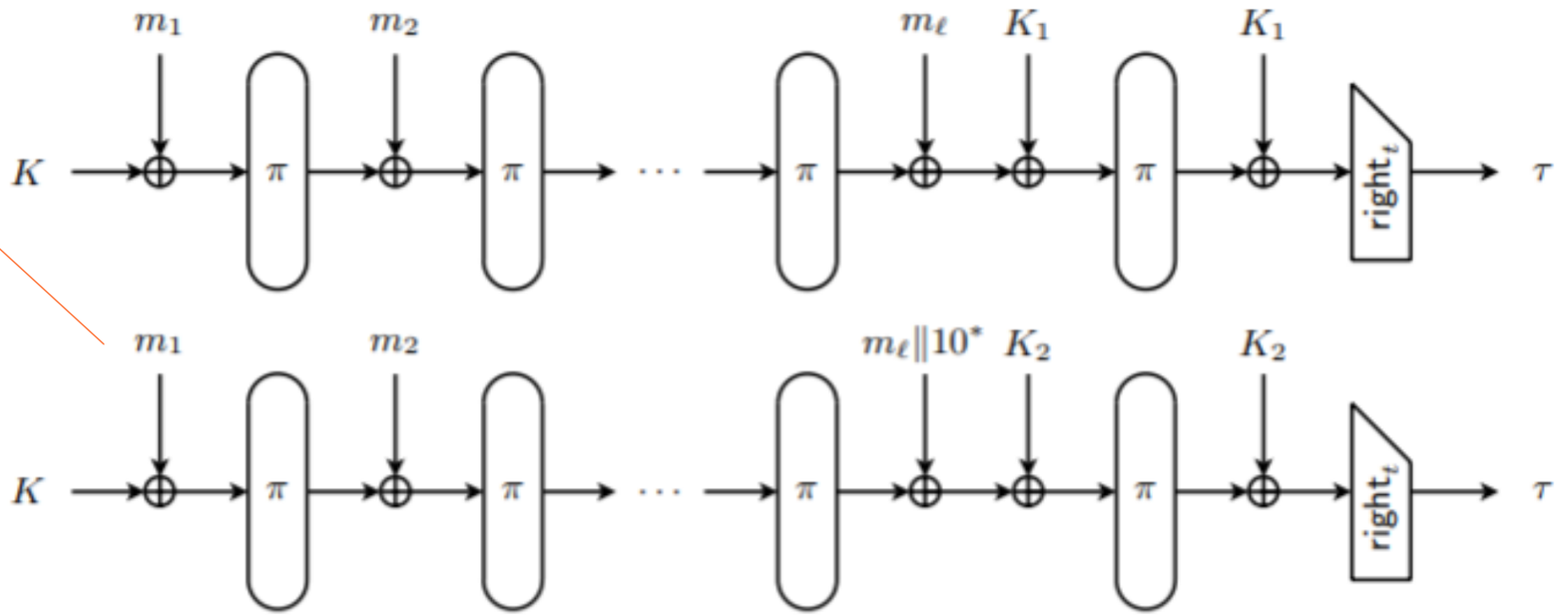
Algorithm 1 TimesTwo

```
1: procedure TimesTwo(a)
2:     if a[127] = 0 then
3:         return (a ≪ 1) ⊕ 0^128
4:     else
5:         return (a ≪ 1) ⊕ 0^120 10000111
```

Algorithm 2 SubKeys

```
1: procedure SubKeys(K)
2:     K1 ← TimesTwo(K)
3:     K2 ← TimesTwo(K1)
4:     return (K1, K2)
```

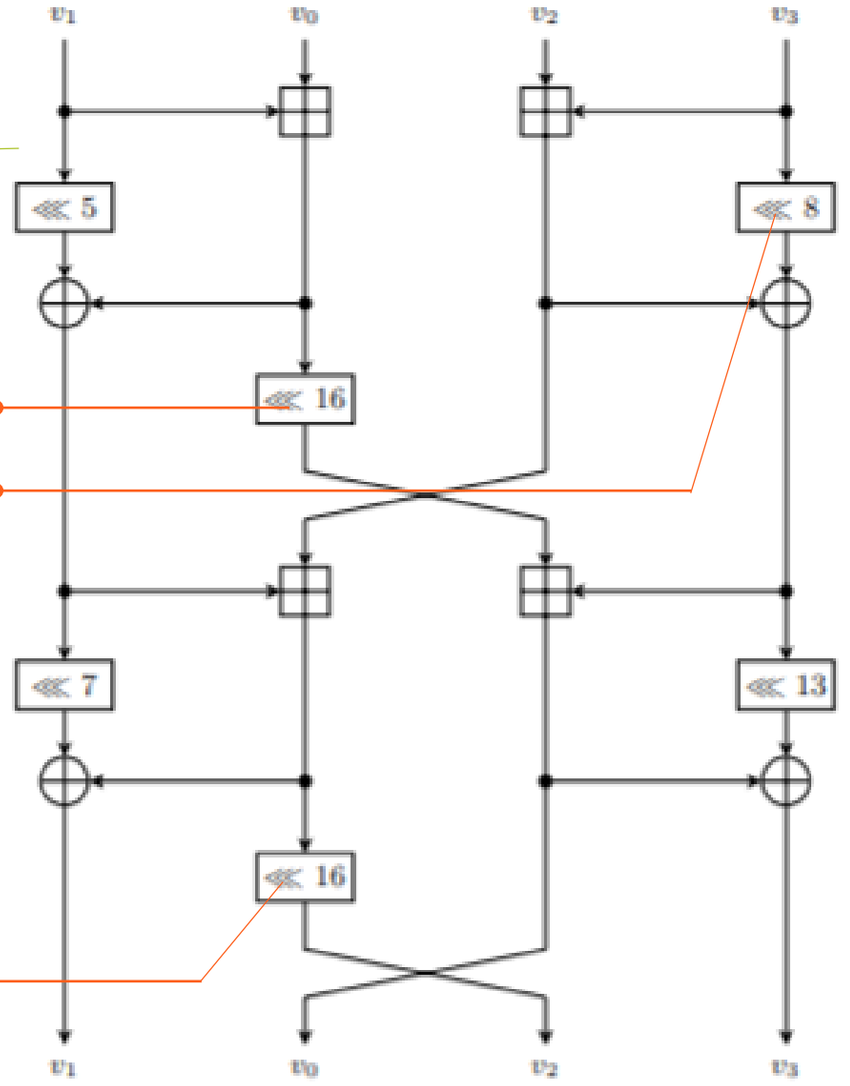
消息M分成大小为n比特的块。然后我们以每n比特为单位对消息进行置换π的迭代，如下图所示。在最后一次的置换π的前后都使用了相同的子密钥进行异或操作，这使得子密钥可以一直保留在寄存器当中，减少装载（load）和存储（store）操作的次数。



Chaskey算法流程

Chaskey算法的置换π是由加法、位循环移位和异或操作构成的。如下图所示，每一轮的置换π由八个操作过程组成。

在一轮的置换π中有一半的位的移位位数都被规定为8的倍数，这是因为有些微控制器并不支持任意位长度的位旋转和平移。由于选择了特殊的常数，这使得Chaskey在8位和16位的微控制器比使用随机位移位位数的算法具有更高的效率。



密码算法介绍——Simon & Speck

Simon和Speck密码是一种轻量级的分组密码，由美国国家安全局在2013年发布，主要为了实现平台的通用性

与其他轻量级分组密码算法不同， Simon和Speck提供了多种分组长度以及每种分组可对应的密钥长度来实现灵活性

Simon和Speck没有使用S盒，它采用的是Feistel结构，可以在线性扩散和非线性混淆的操作之间很好的平衡。

在扩散过程中，为了让密文中的每一位受明文中尽可能多的位影响，需要采用位置换。Simon和Speck算法通过循环移位来实现位置换，在软件和硬件都达到良好的效果。

Simon对于硬件进行优化，Speck对软件进行优化

Simon & Speck算法流程

Simon算法的轮函数为：

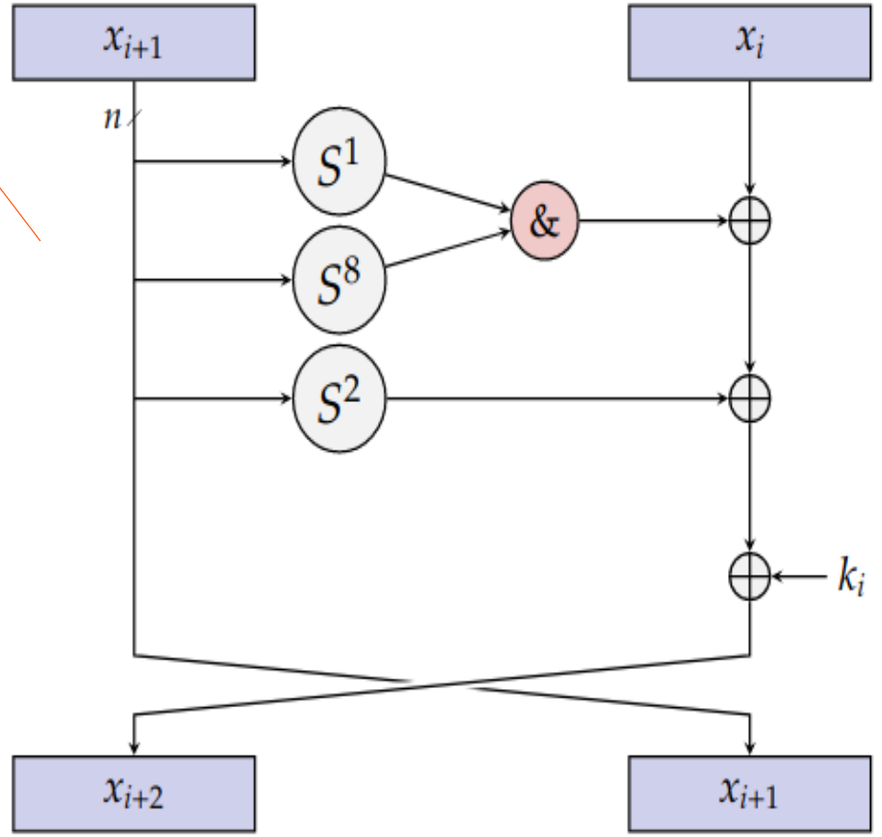
$(x,y)\mapsto(y\oplus f(x)\oplus k,x)$

其中

$f(x)=(Sx\,\&\,S^8x)\oplus S^2x$

S为左循环移位，上标表示移位的位数。

轮函数如右图所示。



根据密钥大小与块大小之间的关系，可将所有情况分为三类：（块大小/密钥长度）2倍（96/96，128/128），3倍（48/72, 64/96, 96/144, 128/192），4倍（32/64, 48/96, 64/128, 128/256）。这三类情况下，轮密钥的产生公式如右图。其中，C、D、E都为常数，I是一个特殊的n×n矩阵，维度等于块大小。

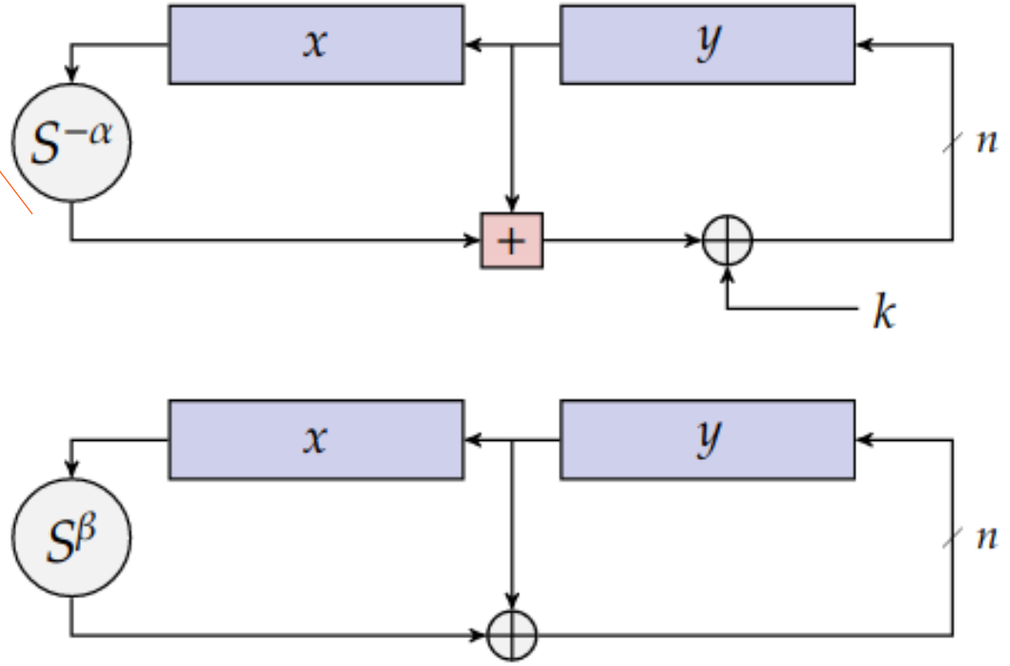
$$k_{i+2} = k_i \oplus (I \oplus S^{-1})S^{-3}k_{i+1} \oplus C_i$$
$$k_{i+3} = k_i \oplus (I \oplus S^{-1})S^{-3}k_{i+2} \oplus D_i$$
$$k_{i+4} = k_i \oplus (I \oplus S^{-1})(S^{-3}k_{i+3} \oplus k_{i+1}) \oplus E_i$$

Simon & Speck算法流程

Speck算法的轮函数为：

$(x,y)\mapsto((S^{-\alpha}\ x+y)\oplus k,\ S^{-\beta}\ y\oplus(S^{-\alpha}\ x+y)\oplus k)$

其中x和y的映射过程可以看做两个Feistel映射的组合，如右图所示。



Speck算法的轮密钥生成函数是基于轮函数得来的，公式如右图所示：

$$\ell_{i+m-1} = (k_i + S^{-\alpha}\ell_i) \oplus i$$
$$k_{i+1} = S^{\beta}k_i \oplus \ell_{i+m-1}$$

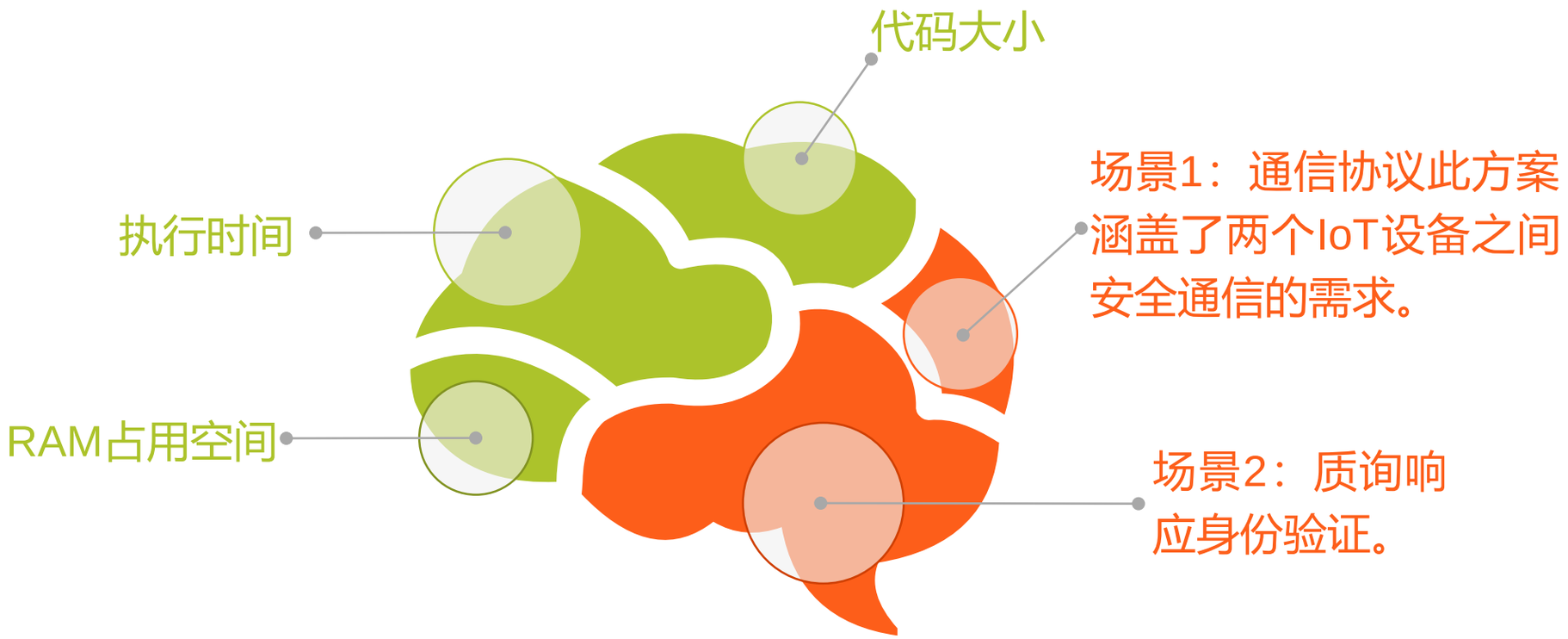
03

PART 03

第三部分

密码性能分析

算法性能测试：三个指标、三个平台与两个场景



文章介绍了一个基准测试框架，用于对众多嵌入式平台的轻量级分组密码进行测试。测试的平台为三个微控制器平台，分别为8位AVR，16位MSP430和32位ARM。

算法性能测试：实验数据

Table 1. Overview of the 1
key sizes are expressed in bit
key setting to the total numb

Cipher	Year			
AES	1998			
Chaskey	2014			
Fantomas	2014			
HIGHT	2006			
LBlock	2011			
LEA	2013			
LED	2011			
Piccolo	2011			
PRESENT	2007			
PRIDE	2014			
PRINCE	2012			
RC5*	1994			
RECTANGLE	2015			
RoadRunneR	2015			
Robin/Robin*	2014			
Simon	2013			
SPARX	2016			
Speck	2013	64	96/128	832/864
TWINE	2011	64	80	1152

* We use RC5 with increased number of rounds, RC5-20.

Table 2. Results for Scenario 1: Encryption and decryption of 128 bytes of data
each platform, the results of the implementation with the best performance indi
shown. The Figure-of-Merit (FOM) is based on the performance indicators on
FOM value, the better the implementations of a cipher).

Cipher	Block [b]	Key [b]	AVR Code [B]	AVR RAM [B]	AVR Time [cyc.]	MSP Code [B]	MSP RAM [B]	MSP Tim
Chaskey	128	12						
Chaskey-LTS	128	12						
Speck	64	9						
Speck	64	12						
Simon	64	9						
Simon	64	12						
RECTANGLE	64	8						
RECTANGLE	64	12						
LEA	128	12						
SPARX	64	12						
SPARX	128	12						
HIGHT	64	12						
AES	128	12						
Fantomas	128	12						
Robin	128	12						
Robin*	128	12						
RC5-20	64	12						
PRIDE	64	12						
RoadRunneR	64	8						
RoadRunneR	64	12						
LBlock	64	8						
PRESENT	64	8						
PRINCE	64	12						
Piccolo	64	8						
TWINE	64	8						
LED	64	8						

* Results for Assembly imple

Table 3. Results for Scenario 1: Encryption a
each platform, the results of the implementatio
shown, whereby different weights were assigne
have a weight of 1, while the execution time h
important than the other two metrics. The F
three platforms (the smaller the FOM value, t

Cipher	Block [b]	Key [b]	AV Code [B]	AV RAM [B]				
Chaskey	128	128	1328*	229				
Chaskey-LTS	128	128	1328*	229				
Speck	64	96	**966***	294				
Speck	64	128	1112*	302				
Simon	64	96	1084*	363				
Simon	64	128	1122*	375				
LEA	128	128	1684*	631				
RECTANGLE	64	80	1152*	352				
RECTANGLE	64	128	1118*	353				
SPARX	64	128	1426*	392				
SPARX	128	128	1736*	753				
AES	128	128	3010*	408				
HIGHT	64	128	1414*	333*	94557*	1258*	330*	120215*
Fantomas	128	128	5892	267	111677	4164	234	56788
Robin	128	128	4944	271	146149	3170	238	76588
Robin*	128	128	5076	271	157205	3312	238	88804
RC5-20	64	128	3706	368	252368	1240	378	386026
PRIDE	64	128	3384	373	111155	2918	380	226135
RoadRunneR	64	80	2504	330	144071	3088	338	235317
RoadRunneR	64	128	2316	**209**	125635	2952	362	218909
LBlock	64	80	2954	494	183324	1632	324	263778
PRESENT	64	80	2160*	448*	245232*	1838*	450*	201549*
PRINCE	64	128	5358	374	243396	4174	240	357423
TWINE	64	80	4236	646	297265	3796	564	387562
Piccolo	64	80	1992	314	407269	1354	310	324221
LED	64	80	5156	574	2221555	7004	252	2065695

* Results for Assembly implementations.

Table 4. Results for Scenario 2: Encryption of 128 bits of data (pre-computed round keys). For each cipher and each platform, the results of the implementation with the best performance indicator according to Equation (1) are shown. The Figure-of-Merit (FOM) is based on the performance indicators on all three platforms (the smaller the FOM value, the better the implementations of a cipher).

Cipher	Block [b]	Key [b]	AVR Code [B]	AVR RAM [B]	AVR Time [cyc.]	MSP Code [B]	MSP RAM [B]	MSP Time [cyc.]	ARM Code [B]	ARM RAM [B]	ARM Time [cyc.]	FOM
Chaskey	128	128	624*	80*	**1465***	388*	70*	**1153***	**216***	76*	**524***	4.4
Chaskey-LTS	128	128	624*	80*	2265*	390*	70*	1690*	**216***	76*	648*	5.0
Speck	64	96	506*	53*	2647*	**328***	**48***	1959*	256	**56**	1003	5.1
Speck	64	128	**452***	53*	2917*	332*	**48***	2013*	276	60	972	5.2
Simon	64	96	600*	57*	4269*	460*	56*	2905*	416	64	1335	7.0
Simon	64	128	608*	57*	4445*	468*	56*	3015*	388	64	1453	7.2
LEA	128	128	906*	80*	4023*	722*	78*	2814*	520*	112*	1171*	8.0
RECTANGLE	64	128	602*									
RECTANGLE	64	80	606*									
SPARX	64	128	662*									
SPARX	128	128	1184*									
RC5-20	64	128	1068									
AES	128	128	1246*									
HIGHT	64	128	636*									
Fantomas	128	128	2496									
Robin	128	128	2530									
Robin*	128	128	2580									
RoadRunneR	64	80	1420									
PRIDE	64	128	2064									
RoadRunneR	64	128	1184									
LBlock	64	80	1440									
PRESENT	64	80	1294*									
PRINCE	64	128	1362									
Piccolo	64	80	1114									
TWINE	64	80	1528									
LED	64	80	2548									

* Results for Assembly implementations.

Table 5. Results for Scenario 2: Encryption of 128 bits of data (pre-computed round keys). For each cipher and each platform, the results of the implementation with the best performance indicator according to Equation (1) are shown, whereby different weights were assigned to the three metrics. Both the memory consumption and code size have a weight of 2, while the execution time has a weight of 1, which means the former two metrics are considered more important than the execution time. The Figure-of-Merit (FOM) is based on the performance indicators on all three platforms (the smaller the FOM value, the better the implementations of a cipher).

Cipher	Block [b]	Key [b]	AVR Code [B]	AVR RAM [B]	AVR Time [cyc.]	MSP Code [B]	MSP RAM [B]	MSP Time [cyc.]	ARM Code [B]	ARM RAM [B]	ARM Time [cyc.]	FOM
Chaskey	128	128	624*	80*	**1465***	388*	70*	**1153***	**184***	76*	**568***	7.1
Speck	64	96	**448***	53*	2829*	**328***	**48***	1959*	256	**56**	1003	7.3
Speck	64	128	452*	53*	2917*	332*	**48***	2013*	264	**56**	1029	7.4
Chaskey-LTS	128	128	624*	80*	2265*	390*	70*	1690*	216*	76*	648*	7.7
Simon	64	96	534*	57*	4521*	460*	56*	2905*	416	64	1335	9.8
Simon	64	128	542*	57*	4709*	468*	56*	3015*	388	64	1453	10.0
RECTANGLE	64	128	602*	56*	4381*	480*	54*	2651*	444*	76*	2365*	11.5
RECTANGLE	64	80	606*	56*	4433*	480*	54*	2651*	444*	76*	2365*	11.5
LEA	128	128	906*	80*	4023*	722*	78*	2814*	520*	112*	1171*	12.1
SPARX	64	128	662*	**51***	4397*	580*	52*	2261*	654*	72*	2338*	12.2
RC5-20	64	128	1068	63	8812	532	60	15925	372	64	1919	18.1
SPARX	128	128	1184*	74*	5478*	904*	80*	3273*	932*	108*	4085*	19.0
HIGHT	64	128	636*	56*	6231*	636*	52*	7117*	670*	100*	5532*	19.7
AES	128	128	1246*	81*	3408*	1170*	80*	4497*	1348*	124*	4044*	21.4
Fantomas	128	128	1712	76	9689	1412	74	5506	1412	104	6484	27.7
Robin	128	128	1712	78	12499	1406	72	7051	1424	116	7686	30.9
PRIDE	64	128	958	60	11222	1842	68	13108	1592	148	7446	33.3
RoadRunneR	64	128	1184	59	6289	756	58	18067	1436	164	8573	33.3
RoadRunneR	64	80	1420	61	7329	1536	76	13034	1900	172	7234	33.7
Robin*	128	128	1754	80	14285	1980	80	5262	1472	116	9186	34.2
LBlock	64	80	1440	64	11183	804	58	16101	616	80	11818	34.7
PRESENT	64	80	1294*	56*	16849*	1072*	58*	12347*	1222*	80*	17105*	44.2
Piccolo	64	80	1114	72	25820	784	70	20081	688	112	17965	48.8
PRINCE	64	128	1362	72	20060	1578	70	24375	1200	132	16270	51.0
TWINE	64	80	1528	64	21701	1922	136	23662	1180	156	15673	52.3
LED	64	80	2602	91	143317	4422	104	121850	2172	352	35891	164.0

* Results for Assembly implementations.

算法性能测试：实验结果

结果表明，最新的ARX和类似ARX的设计使得密码运算不仅非常快，而且在RAM空间占用量和代码大小方面也非常小。以执行时间、RAM占用空间和代码大小为基础的铁人三项比赛的总冠军是Chaskey，紧随其后的是Speck。 两者在两种使用情景下以及在所有三个平台上始终表现良好，这使它们成为保护物联网安全的轻量级密码的强大候选者。当执行时间，RAM占用空间和代码大小权重一样时，Simon密码方案的FOM值也低于10.0，也获得了很好的结果。

04

PART 04

第四部分

总结

总结

通过对三种适用于IoT设备的分组密码Chaskey、Simon和Speck进行分析总结，对它们的设计构造方案和安全性能有了进一步的认识。

通过对它们的性能测试部分的分析，分别得出了在资源受限的IoT设备中，三种密码方案的运行性能情况和对比其他密码方案具有的优势。

在以后的学习和研究带来了一些启发，在设计密码方案的时候，要尽可能考虑在实际运行环境下兼顾安全性和运行性能。

谢谢观看！

Merry Xmas and Happy New Year!!!!

|高亨利 1901210713|刘存展 1901210442|胡兆杰 1901210403|