# UHF RFID 标签的伪随机数发生器研究

高树静[1,2] 王洪君[1]

(山东大学信息科学与工程学院 济南 250000)[1] (青岛科技大学信息科学技术学院 青岛 266001)[2]

**摘 要** 随着物联网的普及,RFID的应用也越来越广泛,对其安全性的研究是近几年的热点。由于成本和计算资源的限制,EPC1类2代(C1G2)无源标签中的安全部件只有随机数发生器和CRC。低复杂性随机数发生器的设计对于C1G2标签的安全是非常关键的。提出一种适于硬件实现的简单哈希函数M-hash,并利用M-hash的单向性设计了一种伪随机数发生器M-PRNG。M-PRNG以LFSR为核心器件,结构简单,适用于C1G2标签等无源器件。经过验证,M-PRNG所产生的随机序列完全符合C1G2协议的要求,并成功通过了NIST测试。

**关键词** 伪随机数发生器,哈希函数,射频识别,EPC1类2代

**中图法分类号** TP309 **文献标识码** A

## Research on PRNG Suitable for UHF RFID Tag

GAO Shu-jing[1,2] WANG Hong-jun[1]

(School of Information Science and Engineering, Shandong University, Jinan 250000, China)[1]

(College of Information Science and Technology, Qingdao University of Science & Technology, Qingdao 266001, China)[2]

**Abstract** With the development of Internet of Things, the application of RFID is becoming more and more prevalent. The security of RFID has been a hot topic in recent years. Due to the limitation of cost and power consumption, the security components in EPC Class 1 Generation 2(C1G2) passive tags are only random number generator (RNG) and Cyclic Redundancy Code (CRC). The design of RNG with low hardware complexity is critical to the security of C1G2 tag. A simple hash function, named M-hash, which is suitable to be realized in hardware was proposed. Furthermore, a pseudo-random number generator M-PRNG was designed based on one-wayness of M-hash. The M-PRNG is based on LFSR and has low hardware complexity which is suitable to passive devices like C1G2 tag. It is proved that the random sequences generated by the M-PRNG are fully compatible with EPC C1G2 protocol and successfully passe the most demanding randomness test NIST.

**Keywords** Pseudo-random number generator, Hash function, RFID, EPC C1G2

## 1 引言

作为条码技术的替代,RFID技术正在被广泛应用。为了推动RFID系统的普及,包括EPCglobal和ISO在内的众多组织都在为RFID标准的制定努力。RFID 1类2代标准[1](本文中简称为C1G2)是由EPCglobal提出来的,并被ISO/IEC接收为国际标准ISO18000-6C。C1G2标签主要用于供应链和仓储管理。作为无源RFID标签,EPC C1G2的主要设计目标是低复杂性,因此协议中的安全考虑很少,标签中仅支持16位的随机数发生器(Random Number Generator, RNG)和16位的循环冗余校验码(Cyclic Redundancy Code, CRC)。由RNG产生的随机数被用于保护口令和防冲突机制的查询过程[1],是保证C1G2标签安全的重要部件。

C1G2协议中并没有对PRNG的结构进行详细的规定,但是根据标签的应用,协议中规定了PRNG必须满足的最低安全标准。

1. 产生单一RN16的概率:从RNG中抽取的RN16等于$j$的概率范围是:

$$\frac{0.8}{2^{16}}<P(j)<\frac{1.25}{2^{16}}$$

2. 同时产生相同序列的概率:对于10000个标签群而言,两个或多个标签同时产生相同RN16序列的概率小于0.1%,无论这些标签何时上电。

3. 预测一个随机数的概率:如果先前从PRNG抽取的随机数已知,在相同条件下从PRNG中抽取的下一个随机数被预测的概率不大于0.025%。

除了上述的最低安全标准之外,作为无源标签,EPC C1G2的低功耗要求也需要满足。因为芯片本身功耗越低,意味着更大的读写距离和更高的系统性能。由于电路功耗对工艺依赖性很强,因此一般以门数来衡量功耗需求。EPC C1G2能够提供用作安全措施的门消耗大约只有400～4k门[2]。

EPC C1G2安全问题的研究是近几年的热点,而针对适用于C1G2标签的RNG的研究并不多,基本上可以分为两类,一类是LFSR结合真随机数的方法,另外一种是伪随机数发生器(Pesudo-Random Number Generator, PRNG)。对于第

到稿日期:2012-09-04 返修日期:2013-01-03 本文受山东省科技攻关项目(2009GG10001007)资助。

高树静(1976－),女,博士生,讲师,主要研究方向为RFID标签芯片设计、RFID系统安全,E-mail:shujing_g@126.com。

一种方法，文献[3]提出了一种基于真随机数(True Random Number，TRN)和 LFSR 的随机数发生器，其可以应用于无源 RFID 标签。作者利用一位由真随机数发生器(True Random Number Generator，TRNG)产生的种子来解决 LFSR 的线性问题，然而其却被证明存在安全隐患[4,5]。文献[4]针对文献[3]的设计进行了改进，设计了多项式可重构的 LFSR，用 TRN 对多项式进行选择。然而其作者并没有详细分析该方法的安全性以及是否符合 C1G2 协议的要求，并且在 RFID 标签中使用 TRNG 还面临着功耗过大的问题。文献[3]中所提出的 TRNG 功耗为 1.04μW，这对于采用类似工艺的 C1G2 标签的 10 μW 功耗预算[6]是比较大的，而标签过多的功耗会影响整个系统的性能。对于第二种方法，Peris-Lopez 等在文献[7]中提出了一种基于基因编程(Genetic Programming，GP)的 PRNG：LAMED，其所产生的随机序列通过了多种随机性测试。然而该方法仍然是基于简单的 XOR 和 OR 等逻辑，非常容易被攻击，并且也没有逐条验证随机序列是否能够满足 EPC C1G2 的所有要求。

因此目前在适用于 EPC C1G2 标签的 PRNG 研究中，还有两个问题需要解决。第一就是低复杂性问题。因为低复杂性意味着低功耗，也意味着更远的阅读距离和更高的系统效率。第二就是要满足协议中的对 PRNG 的最低要求。

本文所提出的基于哈希函数的 M-PRNG 具有更低的硬件复杂性，经过逐条验证，其完全满足协议中的 3 项要求。

哈希函数由于具有单向性而可以被用于 PRNG 的设计[8]，安全 PRNG 的设计的充分必要条件是单向方法的存在[9]。在哈希函数中，基于 LFSR 的哈希函数具有低功耗、低复杂性和高安全性等特点，非常适合于 RFID 标签等无源器件[10]。多输入特征分析寄存器(Multiple-Input-Shift-Register，MISR)的核心结构也是 LFSR，只是其输入采用并行的方式，而不是 LFSR 的串行方式，主要用于集成电路自测试中的测试响应压缩[11]，并且为有损压缩(compaction)，即不可逆的。本文利用这种不可逆特性，提出了一种基于 MISR 的哈希函数(简称为 M-hash)，并进一步以 M-hash 作为单向方法，设计实现了伪随机数发生器 M-PRNG。哈希函数 M-hash 采用并行处理方法，经过两个时钟周期即可产生 16 位的随机数。M-PRNG 采用循环迭代方式，将前一个随机数与标签特有的密钥进行逻辑运算混淆后，作为产生下一个随机数的输入数据，减少了随机数之间的相关性，提高了安全性。经过逐条验证，其安全性完全符合 C1G2 标签的要求。整个电路的规模大约只有 600 门，远远小于 LAMED。此外该发生器所产生的随机序列还通过了权威的随机性测试 NIST，通过比例也高于 LAMED。

本文第 2 节介绍了 M-hash 的构造方法并证明了其冲突概率很低，安全性高；第 3 节设计实现了 M-PRNG，并与 LAMED 进行了比较；第 4 节对 M-PRNG 产生的随机序列进行了测试，通过了最具权威性的 NIST 测试，并将测试结果与 LAMED 进行了比较；第 5 节逐条验证了 M-PRNG 对 C1G2 标签协议的兼容性；最后进行了总结。

## 2 M-hash

### 2.1 M-hash 的构造

MISR 以 LFSR 为核心结构，主要用于集成电路测试的响应压缩[11]。按照 MISR 的位数，将电路的测试响应数据分组，以此输入一组数据，进行循环压缩，将电路的测试响应数据压缩为特征字(signature)，与正确数据相比较，以确定电路是否有故障。MISR 的压缩方式是有损压缩，压缩结果中所包含的输入信息量很少，要想根据压缩结果来恢复输入的测试响应是很难的，符合哈希函数的单向性要求。图 1 给出了基于内部 LFSR 的 MISR。

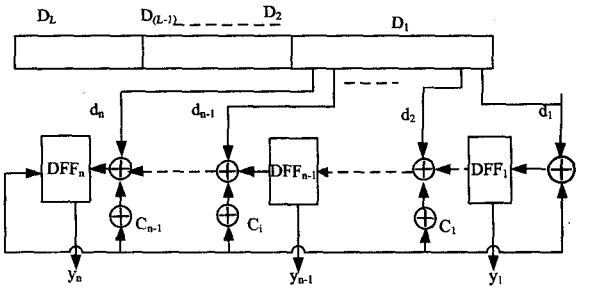


图 1 MISR 哈希

$c_1$ 到 $c_{n-1}$ 是反馈系数，$d_1$ 到 $d_n$ 是并行输入数据。输入向量 $D$ 每个周期被压缩 $n$ 位。$m$ 位输入向量的压缩周期数为 $\lceil m/n \rceil = L$。$D_t$ 表示在第 $t$ 个周期输入的 $n$ 位输入数据。即：

$$D=\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_L \end{bmatrix} \quad D_t=\begin{bmatrix} d_1(t) \\ d_2(t) \\ d_3(t) \\ \vdots \\ d_n(t) \end{bmatrix}$$

设 MISR 的初始状态为 $Y(0)$，$Y(t+1)$ 和 $Y(t)$ 分别表示 MISR 在第 $(t+1)$ 个周期和第 $t$ 个周期的状态。LFSR 的操作可以被描述为 $Y(t+1)=AY(t)$，其中 $A$ 是 LFSR 的转换矩阵。从图 1 可以看出，MISR 的操作可以描述为：

$$Y(t+1)=AY(t)+D(t) \tag{1}$$

式(1)的矩阵表示为：

$$\begin{bmatrix} y_1(t+1) \\ y_2(t+1) \\ y_3(t+1) \\ \vdots \\ y_n(t+1) \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 & \cdots & c_{n-1} & c_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \vdots \\ y_n(t) \end{bmatrix} + \begin{bmatrix} d_1(t) \\ d_2(t) \\ d_3(t) \\ \vdots \\ d_n(t) \end{bmatrix}$$

可以基于式(1)构建哈希函数，为：

$$h(D)=Y(L)=A^L Y(0)+\left(\sum_{j=1}^{L} A^{L-j} D(j)\right) \tag{2}$$

寄存器在第 $L$ 个周期的状态，即 $Y(L)$，就是输入向量 $D$ 的哈希值。

### 2.2 M-hash 冲突概率分析

当与一次性密钥技术相结合提供安全认证时，可以用冲突概率来描述哈希函数的安全性[12]。冲突概率越小，安全性越高。

MISR 的下一个状态只与当前状态和当前输入有关，而与先前状态无关，因此 MISR 行为可以用马尔科夫过程描述。此时马尔科夫过程定义为有限状态机的状态转换。两位多输入特征寄存器及其状态转换如图 2 所示，各个状态表示寄存器状态，弧线上的数据表示状态转换条件，是当前的输入向量 $E=e_1e_2$。设任意时刻任一输入位为 1 的概率是 $p$，为 0 的概率为 $1-p$，则将图 2 中的输入变量用概率代替，就得到了输入信号为随机的情况下 MISR 的马尔科夫过程概率转换图，在此省略。

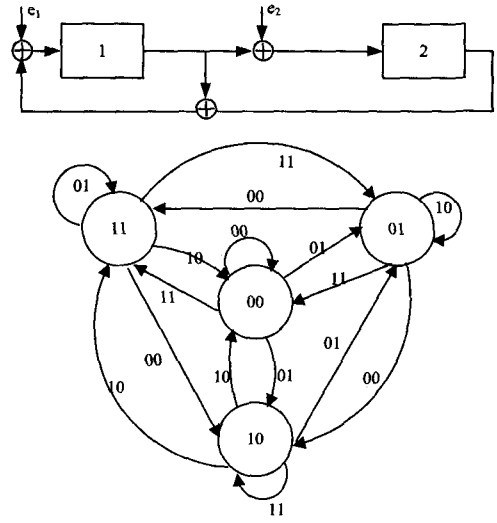


图 2　两位 MISR 的状态转换图

设 $P_{ij}$ 为从状态 $i$ 直接转换到状态 $j$ 的概率，并假设 $P_{ij}$ 为常量。则图 2 所示的马尔科夫概率转换矩阵为：

$$\begin{matrix}(1-p)(1-p) & (1-p)p & p(1-p) & pp\\ pp & p(1-p) & p(1-p) & (1-p)(1-p)\\ p(1-p) & (1-p)(1-p) & pp & p(1-p)\\ p(1-p) & pp & (1-p)(1-p) & p(1-p)\end{matrix}$$

文献[13]证明了当 $n$ 位的 MISR 用于响应压缩时，其冲突概率即错误响应被误当作正确响应的概率为 $\frac{1}{2^n}$。然而该证明忽略了输入向量的长度对冲突概率的影响。

本文将基于下面的假设，在文献[13]已有的结论基础上，给出 MISR 应用于哈希函数时的冲突概率。

假设：输入向量 $E$ 的各个位相互独立，为 1 的概率 $p=\frac{1}{2}$。

对于集成电路测试，由于其各个输入向量之间是有关联的，且各个位之间也不可能相互独立，因此对 aliasing 概率的分析都加入了关于这些相关性的分析[13]。而在哈希函数的应用中，这些错误的引入是由于传输错误或敌手攻击引起的，这种相关性可以不考虑，因此“输入向量 $E$ 各个位相互独立”的假设是恰当的。

**引理 1**　$n$ 位的 MISR 从全零状态出发，输入 $m$ 位向量 $E$ 后回到全零状态的概率为 $P=\frac{(2^n)^{L-1}}{(2^n)^L}$。

证明：根据上面的假设可以得出

$$P=\frac{L\text{ 次状态变换后回到全零初始状态路径数}}{L\text{ 次状态变换的所有可能路径数}}$$

对于当前的任何状态 $X_i$，其下一个可能的状态是所有 $2^n$ 个状态之一，包括其自身，且概率相同。因此，经过 $L$ 次变换之后，MISR 经过的可能路径有 $(2^n)^L$，而其中有 $\frac{1}{2^n}$ 的路径重新回到初始全零状态，即经过 $L$ 次状态变换后回到全零初始状态的路径数是 $(2^n)^L\times\frac{1}{2^n}=(2^n)^{L-1}$，即：

$$P=\frac{L\text{ 次状态变换后回到全零初始状态路径数}}{L\text{ 次状态变换的所有可能路径数}}=\frac{(2^n)^{L-1}}{(2^n)^L}$$

**定理 1**　M-hash(2)的冲突概率为 $P_{colli}<1/2^{-n}$。

证明：假定任意输入向量 $D$ 的哈希值为 $Y(L)$，任意向量 $D_e$ 可以看作是 $D$ 与另外一个变化向量 $E$ 的模 2 加，对应 $D$ 变化的位，相应的 $E$ 为 1。下面计算向量 $D_e$ 的哈希值 $Y_e(L)$：

$$h(D_e)=Y_e(L)=A^LY(0)+(\sum_{j=1}^{L}A^{L-j}D_{ej})$$

$$h(D_e)=A^LY(0)+(\sum_{j=1}^{L}A^{L-j}(D_j+E_j))$$

$$h(D_e)=Y(L)+\sum_{j=1}^{L}A^{L-j}E_j=h(D)+Y_e^*(L)$$

从上式可以看出，若含有变化向量的项 $Y_e^*(L)$ 为 0，则两个哈希值相等，即发生了冲突。

为了验证这种现象出现的概率，不失一般性，可以只考虑 $D$ 的哈希值为全 0 的情形，即 $h(D)=0$。根据这种假设，可以令 $Y_e^*(L)=h(D)+h(D_e)$，则

$$Y_{dL}^*=h(D)+\sum_{j=1}^{L}A^{L-j}E(j)+h(D)=A^L\begin{bmatrix}0\\0\\0\\\vdots\\0\end{bmatrix}+\sum_{j=1}^{L}A^{L-j}E(j)$$

因此，发生冲突的情况可以表示为，状态寄存器初始状态为全零，在以变化向量 $E$ 作为输入数据后，又回到了全零状态。要证明定理 1，只需要证明上述情况发生的概率 $<1/2^{-n}$。

根据引理 1，在输入向量 $E$ 各个位相互独立的前提下，从全零状态回到全零状态的概率为 $P=\frac{(2^n)^{L-1}}{(2^n)^L}$。

回到全零状态的路径中有一条是特殊的，因为在没有错误引入的情形下，MISR 的最后状态也应该回到初始全零状态，当然应该排除在外。因此有：

$$P_{colli}=\frac{(2^n)^{L-1}-1}{(2^n)^L}<1/2^{-n}$$

式中，“−1”即是排除没有错误引入的情形。

结论得证。

## 3　M-PRNG

### 3.1　构造原理

Toeplitz 哈希也是一种基于 LFSR 的哈希函数，其原理是通过 Toeplitz 矩阵与信息向量相乘来获得哈希值。文献[14]采用 Toeplitz 哈希函数作为密钥流发生器设计并实现了流密码发生器。将前一次的随机数与专有密钥进行逻辑运算后作为产生下一次随机数的输入数据，减少了相邻随机数的相关性。在这里将采用文献[14]中的方案构建一个基于 M-hash 的 M-PRNG。

设哈希函数的第 $t$ 个 $m$ 位输入向量为 $Z_t$，输出 $O_t$ 为 $n$ 位的。$O_t$ 即是输入向量 $Z_t$ 的哈希值($m>n$)。$Z_t$ 采取如下的

构建方式：

$$Z_t=\begin{bmatrix}k\oplus O_{t-1}\\ Y_t\end{bmatrix}$$

式中，$O_{t-1}$是第 $t-1$ 个输出，$key$ 是 $n$ 位私有密钥，将 $O_{t-1}$与标签密钥 $key$ 进行异或作为下一个输入向量的一部分。$Y_t$ 是确定性函数 $\phi$ 的第 $t$ 个输出，为 $m-n$ 位的。随机数序列即是哈希函数的连续输出，即：

$O_0=h(key\|Y_0)$

$O_1=h((key\oplus O_0)\|Y_1)$

…

$O_t=h((key\oplus O_{t-1})\|Y_t)$

本文所设计的 M-PRNG 中，哈希函数 $h$ 采用 M-hash，确定性函数 $\phi$ 采用的是具有本原多项式的 $m-n$ 位线性反馈移位寄存器 $LFSR_s$。构建的 M-PRNG 如图 3 所示。

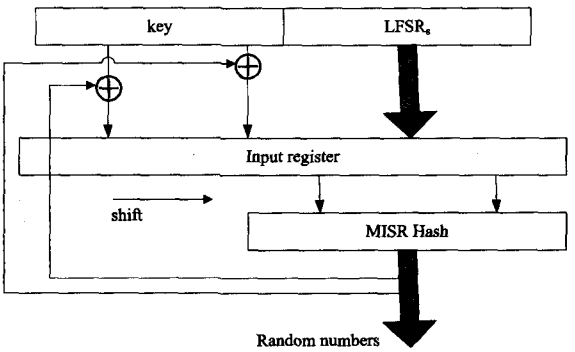


图 3　M-PRNG

M-hash 是 16 位的，密钥 $key$ 和 $LFSR_s$ 都是 16 位的。输入寄存器是 32 位的，用于存储哈希函数的输入数据 $Z_t$。$Z_t$ 被分为两部分：$Z_{1t}$和 $Z_{2t}$，分两次输入到 M-hash 中进行压缩。

$$Z_t=\begin{bmatrix}k\oplus O_{t-1}\\ Y_t\end{bmatrix}=\begin{bmatrix}Z_{1t}\\ Z_{2t}\end{bmatrix}$$

其中 $Z_{1t}=k\oplus O_{t-1}$，$Z_{2t}=Y_t$。

即 $m=32$，$n=16$，每一个周期压缩 16 位输入数据，随机数的产生需要 2 个周期。第一个周期压缩低 16 位数据 $Z_{2t}$，来自 $LFSR_s$，MISR 输出为 $O_{t1}=A_TO_{t-1}+Z_{1t}$。第二个周期压缩高 16 位数据 $Z_{1t}$，来自上一次的哈希值 $O_{t-1}$与密钥 $key$ 的异或。MISR 的输出 $O_{t2}$即为本次输出的随机数 $O_t$。根据式(2)有：

$$O_t=O_{t2}=A_T^2O_{t-1}+A_T(k\oplus O_{t-1})+Y_t \qquad (3)$$

### 3.2　设计实现

由图 3 可知，M-PRNG 的结构简单。MISR 是并行输入的 LFSR，因此总体结构只有两个 16 位的 LFSR、一个 32 位的寄存器、一个 16 位的寄存器和多个异或门。用 verilog 语言设计并实现了 M-PRNG，并以 Altera CYCLONE II EP2C5 为目标进行了综合。结果显示，M-PRNG 的规模仅为大约 600 门，远小于 LAMED[7] 的 1.6k 门，因此在采用相同工艺的前提下，其功耗也远小于 LAMED。在随机数的产生速度方面，LAMED 利用基因编程的原理，用雪崩效应(Avalanche Effect)对产生的随机数进行评估，需要进行反复迭代，周期数不能确定。而 M-PRNG 只需要两个周期就可以产生一个 16 位的随机数，效率更高。在这里假设 RFID 标签数字部分的工作频率为 100kHz[7]，LAMED 和 M-PRNG 产生单个随机数的速度分别为 1.8ms 和 0.02ms。两种 PRNG 的性能比较如表 1 所列。

表 1　两种 PRNG 的性能比较

| | LAMED | M-PRNG |
|---|---|---|
| 硬件消耗 | 约 600 门 | 约 1.6k 门 |
| 单个随机数产生速度(周期数) | 1.8ms | 0.02ms |

## 4　NIST 测试

在本节中，我们将通过随机性测试来证明其安全性。结果表明，与 LAMED 相比，M-PRNG 具有更好的统计特性。

由美国国家标准与技术研究院(NIST)提出的 STS 软件包[15]是最权威的随机性测试工具。在本文中，将使用 STS 系统测试软件(2.1 版)来测试由 M-PRNG 所产生的序列。对于每一个测试，我们计算了 100 个 $p$-$value$，通过的比例如表 2 所列。如果通过比例小于 0.96，则认为测试是失败的。图 4 是将通过比例与 LAMED 进行比较的结果。通过这些结果可以看出，M-PRNG 完全通过了所有的随机性测试，并且通过比例要高于 LAMED，具有更好的随机性。

表 2　NIST 测试结果

| 测试类型 | 通过比例 |
|---|---|
| Frequency | 0.98 |
| BlockFrequency | 1.00 |
| CumulativeSums | 0.98,0.99 |
| Runs | 1.00 |
| LongestRun | 0.99 |
| Rank | 0.97 |
| FFT | 1.00 |
| OverlappingTemplate | 1.00 |
| Universal | 1.00 |
| ApproximateEntropy | 0.99 |
| Serial | 0.99,0.99 |
| LinearComplexity | 0.98 |
| RandomExcursions | 0.98,1.00,1.00,1.00,1.00,1.00,1.00,1.00 |
| RandomExcursionsVariant | 1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00 |

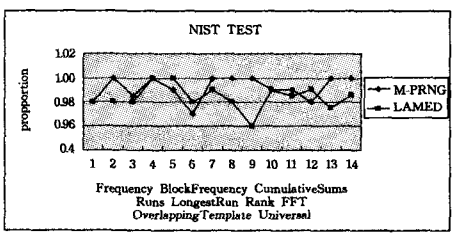


图 4　M-PRNG 和 LAMED 的 NIST 测试比较

## 5　EPC C1G2 标准的兼容性

EPC C1G2 标准定义了 PRNG 必须要满足的 3 个特性。在此将证明，PRNG 完全满足这些要求。

1. 产生单一 RN16 的概率。

我们产生了 10 个文件，每一个文件包含大约 300000 个序列。对每一个值的出现概率进行了统计，结果在表 3 中给出。从表中的数据可以看出，每一个 16 位数据的出现概率范围在$\frac{0.9}{2^{16}}$到$\frac{1.08}{2^{16}}$之间，能够满足标准的要求。

表 3　由基于 MISR 哈希的 PRNG 产生的序列的概率($\frac{p}{2^{16}}$)

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 最低概率 | 0.9235 | 0.9263 | 0.9306 | 0.8997 | 0.9173 | 0.9244 | 0.8845 | 0.9025 | 0.8992 | 0.8829 |
| 最高概率 | 1.0859 | 1.0825 | 1.0788 | 1.0831 | 1.0896 | 1.0903 | 1.0873 | 1.0798 | 1.0867 | 1.0842 |

2. 同时产生相同序列的概率。

我们定义如下的概率。

$P_{collision}$：任意两个或多个标签同时产生相同随机数的概率。根据协议第 2 条的要求，其要小于 0.1%。

$P_{un\text{-}collision}$：所有标签产生的随机序列各不相同。

$P_{same\text{-}in\text{-}diff\text{-}out}$：所有标签中的 MISR 输入数据相同，而输出的哈希值不同。显然此概率为 0。

$P_{diff\text{-}in\text{-}diff\text{-}out}$：所有标签中的 MISR 输入数据不同，而输出哈希值也不同。

$P_{diff\text{-}in\text{-}same\text{-}out}$：所有标签中的 MISR 输入数据不同，而输出哈希值相同。

$P_{diff\text{-}in}$：所有标签的 MISR 输入数据不同。

$P_{MISR\text{-}collision}$：MISR 输入数据不同，而输出数据相同。即 M-hash 的冲突概率。

根据以上的概率定义，可以得出如下的公式：

$P_{collision}=1-P_{un\text{-}collision}$

$P_{un\text{-}collision}=P_{same\text{-}in\text{-}diff\text{-}out}+P_{diff\text{-}in\text{-}diff\text{-}out}=P_{diff\text{-}in\text{-}diff\text{-}out}$

$P_{diff\text{-}in\text{-}diff\text{-}out}=1-P_{diff\text{-}in\text{-}same\text{-}out}$

$P_{diff\text{-}in\text{-}same\text{-}out}=P_{diff\text{-}in}\times P_{MISR\text{-}collision}$，M-hash 的冲突概率为 $P_{colli}<\frac{1}{2^{16}}$。由于 $P_{diff\text{-}in}<1$，因此 $P_{diff\text{-}in\text{-}same\text{-}out}<\frac{1}{2^{16}}$。

$$P_{collision}=1-P_{un\text{-}collision}=1-(1-P_{diff\text{-}in\text{-}same\text{-}out})$$
$$=P_{diff\text{-}in\text{-}same\text{-}out}<0.1\%$$

3. 预测一个随机数的概率。

在式(3)中，假设 16 位的 $Y_t$、种子 $key$ 都是未知的，$A_T$ 是已知的，根据协议第 3 条要求，通过已知的 16 位输出 $O_{t-1}$，推测 $O_t$ 的概率应小于 0.025%。

公式中共有 48 个未知数，因此平均将有 $2^{(48-16)}=2^{32}$ 组不同的解。而其中只有一组解是正确的，因此被预测的概率为 $2^{-32}$，满足协议第 3 条的要求。

**结束语**　安全和隐私问题已经成为 RFID 技术普遍应用的重要障碍。作为 EPC C1G2 的核心安全部件，PRNG 的设计显得尤为重要，并且作为无源标签，只能应用低复杂性的安全措施。本文提出一种低硬件复杂性的哈希函数 M-hash，并基于哈希函数的单向性，提出一种适用于 EPC C1G2 标签的伪随机数发生器 M-PRNG。经过证明，该发生器完全符合协议的要求，并且具备低复杂性的特点，其产生的序列的随机性也通过了权威的 NIST 测试。此外，哈希函数在安全认证协议中被普遍应用[16]，本文在 EPC C1G2 标签中引入低复杂性哈希函数的实现是非常有意义的，为协议安全性的改进提供了很好的借鉴。

## 参 考 文 献


[1] EPCglobal Inc. Class 1 Generation 2 UHF RFID protocol for communication at 860Mhz-960Mhz version 1.0.9

[2] Ranasinghe D, Engels D, Cole P. Low-cost RFID systems: Confronting security and privacy[C]// Auto-ID Labs Research Workshop. 2004

[3] Che W, Deng H, Tan X, et al. A Random Number Generator for Application in RFID Tags[M]. Networked RFID Systems and Lightweight Cryptography, Chapter 16, Springer, 2008: 279-287

[4] Melia-Segui J, Garcia-Alfaro J, Herrera-Joancomarti J. Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags[C]// Sion R, Curtmola R, Dietrich S, eds. RLCPS, WECSR, and WLC 2010. LNCS, vol. 6054, Springer, Heidelberg, 2010: 34-46

[5] Jonan M-S, Joqauin G-A, Jorid H-J. A Practical Implementation Attack on Weak Pseudorandom Number Generator Designs for EPC Gen2 Tags[J]. Wireless Personal Communications, 2011, 59(1)

[6] Pillai V, Heinrich H. An ultra-low-power long range battery / passive RFID tag for UHF and microwave bands with a current consumption of 700 nA at 1.5V[J]. IEEE Transactions on Circuits and Systems I Regular Papers, 2007, 54(7): 1500-1512

[7] Peris-Lopez P, Hernandez-Castro J, Estevez-Tapiador J, et al. LAMED—A PRNG for EPC class-1 generation-2 RFID specification[J]. Computer Standards & Interfaces, 2009, 31(1): 88-97

[8] Menezes A, van Oorschot P, Vanstone S. Handbook of Applied Cryptography[M]. CRC Press, 1996

[9] Goldreich O, Krawczyk H, Luby M. On the existence of pseudorandom generators[C]// IEEE symposium on the foundations of computer science, 29th SFCS. 1988: 12-24

[10] Huang A L, Penzhorn W T. Cryptographic hash functions and low power techniques for embedded hardware[C]// Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005. Vol. 4s, 2005: 1789-1794

[11] Agrawal V D, Kime C R, Saluja K K. A Tutorial on Built-in Self-Test. Part 1: Principles[J]. Design & Test of Computers, IEEE, 1993, 10(1): 73-82

[12] Krawczyk H. LFSR-based hashing and authentication[C]// Advances in Cryptology-crypto'94. Lecture Notes in Computer Science, vol. 839, Springer-Verlag, 1994: 129-39

[13] Min Y, Malaiya Y K, Jin B. Analysis of detection capability of parallel signature analyzers[J]. IEEE Transactions on Computers, 1991, 40(9): 1075-1081

[14] Deepthi P P, Sathidevi P S. Design implementation and analysis of hardware efficient stream ciphers using LFSR based hash functions[J]. Computers & Security, 2009, 28(3/4): 229-241

[15] A statistical test suite for random and pseudorandom number generators for sryptographic applications[OL]. http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/sp800-22rev1a.zip, 2010-04

[16] 丁振华，李锦涛，冯波. 基于 Hash 函数的 RFID 安全认证协议研究[J]. 计算机研究与发展，2009，46(4)：583-592