北京大学学报(自然科学版) 第 53 卷 第 6 期 2017 年 11 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 53, No. 6 (Nov. 2017)
doi: 10.13209/j.0479-8023.2017.059

可用于 SRAM PUF 的密钥提取方案

张亮亮[1,2,†] 张翌维[2] 孙瑞一[2] 周源[2] 王新安[1]

1. 北京大学信息科学技术学院, 北京 100871; 2. 国民技术股份有限公司, 深圳 518057;
† E-mail: zhang.liangliang@nationz.com.cn

摘要 为了利用PUF获得芯片唯一、随机的密钥, 详细分析可用于SRAM PUF的密钥提取方案, 包括采用级联纠错码的硬判决和软判决译码方案。利用芯片上的实际SRAM PUF响应和软件仿真, 验证两种方案的效果。结果表明, 对于SRAM PUF, 软判决方案更加可靠和高效。

关键词 物理不可克隆函数(PUF); 密钥提取; 级联纠错码; 硬判决纠错; 软判决纠错
中图分类号 TP309

Key Extraction Schemes for SRAM PUF

ZHANG Liangliang[1,2,†], ZHANG Yiwei[2], SUN Ruiyi[2], ZHOU Yuan[2], WANG Xin'an[1]

1. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871; 2. Nationz Technologies Inc., Shenzhen 518057; † E-mail: zhang.liangliang@nationz.com.cn

Abstract To acquire unique and random keys for chips from PUF, key extraction schemes for SRAM PUF are introduced. These schemes include the hard-decision and soft-decision decoding ones involving concatenated error-correcting code. By the actual responses of SRAM PUF's on chips and software simulation, the effect of above two schemes is verified. The result shows that, for SRAM PUF, the soft-decision scheme is more reliable and efficient.

Key words PUF; key extraction; concatenated error-correcting code; hard-decision error correction; soft-decision error correction

PUF (physical unclonable function)[1–2]指一种物理实体或物理结构, 它易于求值, 但很难复制。PUF 最重要的特性是, 即使知道具体结构和制造方式, 要想复制与某个设备的 PUF 相同的实体或结构, 是很难或几乎不可能实现的。PUF 的特性使得它在密码学和集成电路 (integrate circuit, IC)设计制造领域引起广泛的兴趣, 可用于密码芯片以及射频识别标签等[3–5]。IC 芯片上的 PUF 利用了芯片制造过程中工艺偏差造成的芯片之间内在的随机差异, 可视为芯片或设备的"指纹", 用于芯片的身份识别和认证。硅芯片上的 PUF 称为硅 PUF, 可以很方便地集成到通常的芯片中。常见的两类硅 PUF 包括基于数字电路回路延时的 PUF[6–8]以及基于存储单元上电初始值的PUF[4,9–11]。SRAM (static random access memory) PUF 就是利用了 SRAM 存储单元上电后初始值是随机的这一特性[1]。

PUF的结构如图 1 所示, 它可以接受激励(challenge)作为输入, 经过求值得到一个响应(response)作为输出。例如在 SRAM PUF 中, SRAM 的地址可以视为激励, 上电初始值作为响应。PUF 可以每次输入相同的激励, 然后将输出的响应作为密钥, 这个密钥不会像通常地那样存储在非易失存储器中, 而是只在需要时在芯片内部生成, 并且不离开芯片, 减少了被攻击的风险, 可以作为一种更安全的密钥存储方式。但是, 即使对于同一个激励, 由于 PUF 内部噪声等原因, 不同时候得到的响应不相同。

国家核高基科技重大专项(2014ZX01032-204, 2014ZX01032401-001)和深圳市重点实验室基金(深发改[2013]993号)资助
收稿日期: 2016–06–27; 修回日期: 2016–12–26; 网络出版日期: 2017–02–28

图 1 PUF 示意图
Fig. 1 Sketch map of PUF

为了得到唯一不变的密钥, 需要对 PUF 的响应进行处理。本文介绍的密钥提取方案可以保证, 若 PUF 响应在一定范围内变化, 利用纠错码可以恢复注册阶段使用的密钥, 可以确保获取的密钥的唯一性。本文利用实际芯片中 SRAM 的上电初始值进行测试, 验证密钥提取方案的有效性。

1 密钥提取方案

实现某种类型 PUF 的不同芯片称为 PUF 不同的实例。不失一般性, PUF 的激励和响应都可以视为二进制的比特串。不同的 PUF 实例对相同的激励会产生不同输出, 这些输出的二进制比特串之间的汉明距离称为片间汉明距离 (μ_{inter}), 若用百分比表示, 片间汉明距离在理想情况下为 50%。某一个 PUF 实例对于相同的激励产生的二进制输出之间的汉明距离称为片内汉明距离 (μ_{intra}), 若用百分比表示, 片内汉明距离在理想情况下为 0%。但在实际情况下, μ_{inter} 和 μ_{intra} 与理想值都有差别, 并且都具有一定的分布。对于实际上可用的 PUF, 应该满足 $\hat{\mu}_{\text{inter}} \gg \hat{\mu}_{\text{intra}}$, $\hat{\mu}$ 为汉明距离在各自分布情况下的均值。

由于 $\hat{\mu}_{\text{intra}} \neq 0\%$, 所以对于相同的激励, PUF 在不同时刻产生的响应会存在比特差异, 密钥提取方案需要处理这些比特差异来生成唯一的密钥。另外, 密钥提取方案生成的密钥应该具有足够的熵, 以保证应用的安全性。

1.1 模糊提取器方案

PUF 的密钥可以通过模糊提取器(fuzzy extractor)进行提取[1,12–13]。模糊提取器通常由两部分组成: 一部分用于恢复一定长度的响应; 另一部分用于数据压缩和熵累加。

模糊提取器中用于恢复长度为 n 比特响应的部分可以通过码字偏移(code-offset)方案来构造。假定有参数为[n, k, t]的线性分组纠错码 $\mathcal{C}$, 其中 k 为消息长度, n 为编码后的码字长度, t 为此码最多能纠正的错误个数。为了纠正多个随机错误, 这些纠错码可以选择 BCH 码、Golay 码以及 Reed-Muller 码等[14–15]。码字偏移方案的过程[16–17]如下。

1) 注册阶段: 随机从纠错码 $\mathcal{C}$ 的所有可用码字中选择一个码字 c; 读取一个长度也为 n 的 PUF 响应, 记为 y, 得出 $w = y \oplus c$。w 称为辅助数据, 可以公开, 不需要保密存储。

2) 恢复阶段: 再读取 PUF 的一个响应, 其激励应该与注册阶段的相同, 此时响应记为 y'。由于 PUF 响应存在片内汉明距离, 所以 y' 和 y 通常不会相同。令 $c' = y' \oplus w$, 将 c' 视为一个经过传输带有误差的码字, 用纠错码进行译码, 得到 c''。恢复的值 $y'' = c'' \oplus w = y \oplus (c \oplus c'')$。可以看出, 如果 c' 与 c 的汉明距离小于 t, 则纠错码可以将 c' 正确地恢复成 c, $y'' = y$。这意味着将注册阶段的响应 y 成功地恢复。若 c' 与 c 的汉明距离大于 t, 则超出纠错码的纠错能力范围, 无法将 c' 正确地恢复成 c, 也就无法恢复出 y。

假定 PUF 响应的最小熵密度为 ρ, 由于 w 公开, 并且码字中独立的比特数目只有 k 个, 因此在已知 w 的情况下, n 比特响应的条件熵为 $\rho n - (n-k) = k - n(1-\rho)$, 这个值大于 0 才有意义。如果一个响应的条件熵不满足期望的要求, 可以使用多个长度为 n 的响应, 使得最后密钥的熵满足要求。另外, 纠错码的纠错能力 t 的设计与 PUF 响应发生的比特错误个数有关。例如, 要设计一个密钥提取方案, 它能成功地恢复熵为 s 比特的密钥的概率不小于 $1-\varepsilon$ (ε 为一个很小的数)。我们假定 PUF 响应的每个比特发生错误的概率是 p, 并且比特错误的产生是彼此独立的, 这样 PUF 长度为 n 的响应中比特错误的个数 e 服从参数为 n, p 的二项式分布 $b(e; n, p)$。生成熵为 s 比特的密钥需要的长度为 n 的响应的个数为 $x = \left\lceil \dfrac{s}{k - n(1-\rho)} \right\rceil$, 则 t 需要满足 $[b(t; n, p)]^x \geqslant 1-\varepsilon$。所以, 若要设计一个 PUF 的密钥提取方案, 需要给出或指定 ρ, ε, p 和 s, 然后根据一些设计准则来找出合适的纠错码[n, k, t]以及需要的 PUF 响应长度 $l = xn$, 这些设计准则为

1) $\dfrac{k}{n} > (1-\rho)$;

2) $x = \left\lceil \dfrac{s}{k - n(1-\rho)} \right\rceil$, $l = xn$;

3) $t \geqslant b^{-1}\left((1-\varepsilon)^{\frac{1}{x}}; n, p \right)$。

把注册阶段使用的 PUF 响应恢复后, 需要通过熵累加器进行数据压缩来获取密钥。直接采用密

码学哈希函数[17–18]即可，例如 SHA、基于线性反馈移位寄存器的 Toeplitz 哈希函数或者一些轻量级的哈希函数等。本文主要关注密钥提取方案中的纠错部分，不对熵累加器进行详细论述。

1.2　采用级联码的密钥提取方案

PUF 响应每个比特发生错误的概率 p 可能会达到10%，甚至更大，所以模糊提取器方案中的纠错码需要很强的纠错能力，这会导致选择的纠错码码字相对较长，纠错速率较低。为了解决这个问题，可以采用级联码进行编码和译码[16–17]。译码时，首先采用一个较简单但具有一定纠错能力的纠错码(如重复码)作为内码，将码字的错误降到相对较低的水平，然后采用纠错能力更强的码作为外码，完成最后的纠错。在典型情况下，内码可以采用重复码 $\mathcal{C}_1[n_{\text{rep}}=2t_{\text{rep}}+1,1,t_{\text{rep}}]$，外码采用 BCH 码等纠错码，记为 $\mathcal{C}_2[n_2,k_2,t_2]$。假定用来生成密钥的 PUF 响应长度为 $l=x\cdot n_{\text{rep}}n_2$。

采用级联码的码字偏移方案如下。

1) 注册阶段。

① 随机选择 x 个 $\mathcal{C}_2[n_2,k_2,t_2]$ 的码字，将它们看作长度为 xn_2 的消息，使用 $\mathcal{C}_1[n_{\text{rep}},1,t_{\text{rep}}]$ 分别对每个消息元进行编码，将它们按顺序合并到一起，得到一个总长度为 l 的比特串 c。

② 采用某个激励读取一个长度为 l 的 PUF 响应 y，计算 $w=y\oplus c$，将 w 作为辅助数据，w 可以公开。

2) 恢复阶段。

① 利用与注册阶段相同的激励读取一个 PUF 的响应 y'，计算 $c'=y'\oplus w$，这时 c' 相当于一个有传输比特错误的码字组合。

② 将 c' 分成 xn_2 个长度为 n_{rep} 的码字，用内码 $\mathcal{C}_1[n_{\text{rep}},1,t_{\text{rep}}]$ 对其分别进行译码，错误个数小于等于 t_{rep} 的被纠正，错误个数超过 t_{rep} 的则被错误地译码，可以得出长度为 xn_2 的消息。这个消息可以视为 x 个 $\mathcal{C}_2[n_2,k_2,t_2]$ 经过传输的带有比特错误的码字。

③ 将这 x 个 $\mathcal{C}_2[n_2,k_2,t_2]$ 的码字用外码分别进行译码，可以译得 x 个码字。

④ 将译得的 x 个 $\mathcal{C}_2[n_2,k_2,t_2]$ 码字视为长度为 xn_2 的消息，用内码 $\mathcal{C}_1[n_{\text{rep}},1,t_{\text{rep}}]$ 再次进行编码，采用与注册阶段步骤①相同的方式，将它们按顺序合并到一起，这相当于恢复了一个长度为 l 的 c''。

⑤ 计算 $y''=w\oplus c''$。如果在整个译码过程中错误被全部纠正，则 $c''=c$，从而 $y''=y$。否则，注册阶段的响应恢复失败。

对于某个 PUF，假定它的响应熵密度为 ρ，响应的每个比特位发生比特错误的概率为 p，且不同比特位的错误是独立出现的。要以不小于 $1-\varepsilon$ 的概率生成一个熵为 s 的密钥，使用级联码纠错方案的密钥提取方案时，需要确定纠错码 $\mathcal{C}_1[n_{\text{rep}}=2t_{\text{rep}}+1,1,t_{\text{rep}}]$ 和 $\mathcal{C}_2[n_2,k_2,t_2]$ 的参数以及需要的 PUF 响应长度 $l=x\cdot n_{\text{rep}}n_2$。这些参数需要满足的约束关系为

$$1)\ \frac{k_2}{n_{\text{rep}}n_2}>1-\rho;$$

$$2)\ x=\left\lceil\frac{s}{k_2-n_{\text{rep}}n_2(1-\rho)}\right\rceil,\ l=x\cdot n_{\text{rep}}n_2;$$

$$3)\ t_2\geqslant b^{-1}\left(\left(1-\varepsilon\right)^{\frac{1}{x}};n_2,p'\right),\ p'=1-b(t_{\text{rep}};n_{\text{rep}},p)。$$

1.3　软判决纠错方案

只要比特错误的个数没有超过纠错码最大可以纠正的错误个数，模糊提取器方案和采用级联码的密钥提取方案就可以纠正任意位置发生的比特错误，这种方案称为硬判决译码方案。显然，硬判决译码方案对于任何种类的 PUF 都是可行的。硬判决译码方案认为每个比特发生错误的概率都是相同的 p，p 可以通过响应中比特错误的平均数估计[1]。

SRAM PUF 的上电初值有如下特性[19]：大部分响应比特十分稳定，不同时刻的上电初值很少，或者几乎不改变；只有少部分响应比特不稳定，不同时刻的上电初值经常不同。这说明对于 SRAM PUF 响应，每个比特有各自不同的比特错误率。所以，更有效的密钥提取方案应该考虑并利用 PUF 响应中每个比特的特性，比如提前测试 PUF 响应中的每个比特的误差概率(这些信息可以公开)，然后利用这些信息进行判决译码，这种方式称为软判决译码。由于有额外的信息辅助进行译码，所以软判据译码通常会更加高效[18–19]。

假定 SRAM PUF 的 n 比特响应中，每个比特的错误率为 $p_i(i=1,...,n)$，依照硬判决译码方案的码字偏移方案，这些错误率会反映为码字 c 中每个码元 $c_i(i=1,...,n)$ 的传输错误概率。设经过传输后的码字为 c'，码元为 $c_i'(i=1,...,n)$。软判决的纠错原则就是找到某个码字 c^*，使得下面的对数似然估计量最大:

$$\sum_{i=1}^{n}(-1)^{c_i'+c_i^*}\log_\beta\frac{1-p_i}{p_i},$$

其中$\beta>1$。

SDML (soft-decision maximum-likelihood decoding)译码方案选择使上式的对数似然估计量最大的码字。由于 SDML 方案需要遍历所有的 2^k 个码字，因此 SDML 方案的译码效率与纠错码的维度 k 呈指数依赖关系。但是，对于重复码(k=1)，SDML 可以有效地译码。令 $P_i=\log_\beta\frac{1-p_i}{p_i}$，对于重复码[$n$, k, t]，计算 $L_i=(-1)^{c_i'}\cdot P_i$，$L^*=\sum_{i=1}^{n}L_i$。若 $L^*>0$，则传输的码字 c 为全 0 码字的可能性最大；若 $L^*<0$，则 c 最可能为全 1 码字。选择可能性最大的码字作为译码结果，SDML 就完成对重复码的译码。

r 阶 Reed-Muller 码 $\mathrm{RM}(r,m)$ 是一种具有良好结构的线性纠错码[14]，$n=2^m$，$k=\sum_{i=0}^{r}\binom{m}{i}$，$t=\left\lfloor\frac{2^{m-r}-1}{2}\right\rfloor$。RM($r$, m)可以由 RM(r−1, m−1)和 RM(r, m−1)两个较短的码组合而成。反复进行这种分解，直到 RM(0, m)或者 RM(r, r)。RM(0, m)是一个简单的重复码，而 RM(r, r)是退化的编码，可以认为其效果是没有进行编码。显然，RM(0, m)和 RM(r, r)都可以用 SDML 进行译码。这种技术称为广义多重级联解码(generalized multiple concatenated decoding, GMC)[20]。

采用级联码的密钥提取方案，将 RM(r, m)与重复码进行级联，这样能同时利用软判决解码和级联码的优点。类似 1.2 节的步骤，注册阶段生成级联码的码字，并与 PUF 响应进行异或生成辅助数据。在恢复阶段，首先用内码重复码的 SDML 译码方案进行译码，然后用外码 Reed-Muller 码 RM(r, m)的 GMC 译码方案进行译码，需要将其中重复码的输出 L^*组合起来直接作为 GMC 译码方案的输入。译码过程需要利用软判决 SDML 算法和 RM 码软判决 GMC 算法。

软判决 SDML 算法。

SDML_DECODE_Rep (L, n)

$L^*=\sum_{i=1}^{n}L_i$

return $(L^*,\cdots,L^*)_n$

RM 码软判决 GMC 算法。

GMC_DECODE_RM(L, r, m)

$F(a,b):=\mathrm{sign}(a\cdot b)\cdot\min(|a|,|b|)$

$G(s,a,b):=\left\lfloor\frac{1}{2}(\mathrm{sign}(s)\cdot a+b)\right\rfloor$

if r= 0

  L^*= SDML_DECODE_Rep(L, 2^m)

else if $r=m$

  $L^*=L$

else

  $L_i^{(1)}=F(L_{2i-1},L_{2i})$, $i=1,\ \ldots,\ 2^{m-1}$

  $L^{(1)*}$= GMC_DECODE_RM ($L^{(1)}$, r−1, m−1)

  $L_i^{(2)}=G(L_i^{(1)*},L_{2i-1},L_{2i})$, $i=1,\ \ldots,\ 2^{m-1}$

  $L^{(2)*}$= GMC_DECODE_RM($L^{(2)}$, r, m−1)

  $L^*=(F(L_1^{(1)*},L_1^{(2)*}),L_1^{(2)*},\ldots,F(L_{2^{m-1}}^{(1)*},L_{2^{m-1}}^{(2)*}),L_{2^{m-1}}^{(2)*})$

end

return $(L^*,\ldots,L^*)_n$

2 仿真结果与分析

2.1 硬判决密钥提取方案仿真结果

我们利用某款 90 nm 工艺的芯片，对硬判决密钥提取方案进行仿真。首先，对芯片多个样片的 SRAM 上电初始值进行统计，发现片内距离的均值 $\hat{\mu}_{\mathrm{intra}}=15.70\%$，它可以作为响应的每个比特位发生比特错误的概率 p，即 $p=15.70\%$。假定响应的熵密度 $\rho=95\%$，本文设计一个密钥提取方案，可以以不小于 $1-10^{-6}$ 的概率恢复熵为 $s=128$ 比特的密钥，仿真结果如表 1 所示。

按照采用级联码的密钥提取方案设计准则，找到的纠错码组合为表 1 中的方案 1：将重复码 $\mathcal{C}_1[7,1,3]$ 与缩短的二进制 BCH 码 $\mathcal{C}_2[488,299,22]$

表 1 硬判决密钥提取方案的仿真结果

Table 1 Simulation results for the key extraction schemes using hard-decision algorithms

方案	内码	外码	x	l	实际失败概率
1	$\mathcal{C}_{\mathrm{rep}}[7,1,3]$	$\mathcal{C}_{\mathrm{BCH}}[488,299,22]$	1	3416	0.16
2	$\mathcal{C}_{\mathrm{rep}}[7,1,3]$	$\mathcal{C}_{\mathrm{BCH}}[1754,742,105]$	1	12278	5×10^{-5}

进行级联，并且 $x=1$。对于芯片实际的 PUF 响应，进行1000 次密钥提取，发现失败160 次，失败率远远超出设计要求的10^{-6}。经过检查，发现无法成功进行密钥恢复的响应中，重复码没有正确译码的消息个数很多，导致比特错误个数超出 BCH 码的纠错能力范围。理论上，对于重复码 $\mathcal{C}_1[7,1,3]$，无法正确纠正比特错误的概率为 $1-b(3;7,15.70\%)=1.43\%$；实际上，方案 1 中长度为 $l=3416$ 的 PUF 响应中，每 7 个比特发生 3 个以上比特错误的比例全部高于1.43%，其中最多的为 6.56%。这说明硬判决采用的假设与实际偏差较大，即 PUF 响应中“每个比特发生错误的概率是相同的，并且是彼此独立的”这个假设与实际情况差别较大。

表 1 中的方案 2 采用重复码 $\mathcal{C}_1[7,1,3]$和缩短的 BCH 码 $\mathcal{C}_2[1754,742,105]$ 级联码的硬判决方案，经过 10 万次测试，失败概率为 5×10^{-5}。此方案提高了级联码的纠错能力，即使发生较多的比特错误时，也能正确地提取密钥，但这种简单地提高级联码纠错能力的方法会显著增加提取方案的实现代价。与方案 1 相比，方案 2 需要的 PUF 响应长度更长，且 BCH 码的参数更大，从而导致硬件 IP 的面积更大，所需的时钟周期数也更长。

2.2　软判决密钥提取方案仿真结果

进行软判决密钥提取方案前，需要将 PUF 响应读取多次(本文选择 100 次)，然后计算每个比特的错误概率 p_i，并保存。软判决译码方案中的最佳参数可以根据实际测试结果来确定。仿真结果如表 2 所示，这些方案的实际失败概率都利用相同的 PUF 响应，进行 10 万次测试而得到。

表 2 中，方案 1 和方案 2 采用级联码的方案。比较两个方案可以发现，在内码重复码固定为 $\mathcal{C}_{\text{rep}}[3,1,1]$ 的情况下，与 RM(2, 6)相比，RM(2, 7)码字之间的距离更长，纠错能力更强，所以方案 2 的纠错效果几乎提高一个量级。$\mathcal{C}_{\text{rep}}[3,1,1]$ 和 RM(2, 7)级联的软判决方案是所有测试中密钥提取效果最好的纠错码组合。

为了比较采用级联纠错码与单独采用某一种纠错码方案的效果，我们仿真仅仅采用 RM(2, 8)和 RM(2, 9)的方案(表 2 中方案 3 和 4)。通过与方案 2 比较可以发现，RM(2, 7)和 $\mathcal{C}_{\text{rep}}[3,1,1]$级联方案的效果可以达到单独采用 RM(2, 9)方案的效果，所以在码的参数较短情况下，采用级联码方案可以达到更好的恢复效果。另外，经过测试发现，$\beta=1.6$ 时，恢复效果比较好，表 2 中的方案都选择这个值。

与硬判决方案相比，软判决方案使用的纠错码的码长等参数更短，因此效率以及硬件面积开销也更小[17−18]。软判决方案的代价是需要提前测量并计算每个比特的错误概率 p_i，然后利用它们计算得到许多对数值，并将这些对数值保存起来，这需要额外的操作和存储空间。通常，需要读取几十次每个 SRAM 单元的上电初始值，并计算 p_i，然后计算 $P_i=\log_\beta\frac{1-p_i}{p_i}$，并将 P_i 存储为 16 比特长度的带符号的整数[18]。本文最好的方案为表 2 中方案 2，它需要的存储空间约为 11 KBytes。如果能接受这些代价，软判决方案将比硬判决方案更加快速和可靠。

3　总结

通过利用芯片内在的差异，PUF 可以达到识别认证的目的，也可以作为保护密钥的一种更安全的方案，因此 PUF 是一种物理的可信根(root of trust)。密钥提取方案的基本思想是将 PUF 响应的比特误差转换成纠错码码字的比特误差，然后利用纠错码的译码算法，恢复注册阶段使用的密钥，这样就可以每次都恢复唯一的密钥。为了提高密钥提取方案的效率，可以采用级联码的纠错码方案以及利用已知的一些信息的软判决方案。本文将采用级联码的硬判决方案和软判决方案用于某款芯片的 SRAM

表 2　软判决密钥提取方案的仿真结果

Table2　Simulation results for the key extraction schemes using soft-decision algorithms

方案	内码	外码	x	l	实际失败概率
1	$\mathcal{C}_{\text{rep}}[3,1,1]$	RM(2, 6) = [64, 22, 7]	11	2112	1.6×10^{-4}
2	$\mathcal{C}_{\text{rep}}[3,1,1]$	RM(2, 7) = [128, 29, 15]	14	5376	3×10^{-5}
3	无	RM(2, 8) = [256, 37, 31]	6	1536	4×10^{-5}
4	无	RM(2, 9) = [512, 46, 63]	7	3584	3×10^{-5}

PUF，设计了几种方案的参数，通过软件仿真比较密钥提取的实际效果，证明若使用级联码的软判决方案，纠错码参数会更短，并且密钥提取效果更好。本文只关注正常工作条件下的 PUF 响应，实际使用时需要讨论温度、电压等对 PUF 的影响[21]。另外，如何将这些密钥提取方案安全、高效地用硬件 IP 实现，仍是一个值得研究的问题。

参考文献

[1] Maes R. Physically unclonable functions: constructions, properties and applications. Berlin: Springer-Verlag Berlin Heidelberg, 2013

[2] Böhm C, Hofer M. Physical unclonable functions in theory and practice. New York: Springer, 2013

[3] 恩智浦以物理不可克隆技术(PUF)强化 SmartMX2 安全芯片. 微电脑世界, 2013(4): 17

[4] Holcomb D E, Burleson W P, Fu K. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags // Workshop on RFID Security and Privacy: RFIDSec 2007. New York: IEEE, 2007

[5] 寇红召, 张紫楠, 马骏. 基于物理不可克隆函数的 RFID 双向认证. 计算机工程, 2013, 39(6): 142–145

[6] Gassend B, Clarke D, van Dijk M, et al. Silicon physical random functions // CCS 2002: Proceedings of the 9th ACM Conference on Computer and Communications Security. New York: ACM, 2002: 148–160

[7] Gassend B, Lim D, Clarke D, et al. Identification and authentication of integrated circuits: research articles. Concurrency and Computation: Practice and Experience, 2004, 16(11): 1077–1098

[8] Lee J W, Lim D, Gassend B, et al. A technique to build a secret key in integrated circuits for identification and authentication applications // Symposium on VLSI Circuits-VLSIC 2004. New York: IEEE, 2004: 176–179

[9] Guajardo J, Kumar S S, Schrijen G J, et al. FPGA intrinsic PUFs and their use for IP protection // Workshop on Cryptographic Hardware and Embedded Systems-CHES 2007. Berlin: Springer, 2007: 63–80

[10] Holcomb D E, Burleson W P, Fu K. Power-up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Transactions on Computers, 2009, 58(9): 1198–1210

[11] Schrijen G J, van der Leest V. Comparative analysis of SRAM memories used as PUF primitives // Design, Automation and Test in Europe-DATE 2012. New York: IEEE, 2012: 1319–1324

[12] Dodis Y, Reyzin L. Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data // Advances in Cryptology-EUROCRYPT 2004. Berlin: Springer, 2004: 523–540

[13] Dodis Y, Ostrovsky R, Reyzin L, et al. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM Journal on Computing, 2008, 38(1): 97–139

[14] Lin S, Costello D J. Error control coding. 2nd ed. New Jersey: Pearson Prentice-Hall, 2004

[15] 王新梅, 肖国镇. 纠错码——原理与方法(修订版). 西安: 西安电子科技大学出版社, 2001

[16] Bösch C, Guajardo J, Sadeghi A R, et al. Efficient helper data key extractor on FPGAs // Workshop on Cryptographic Hardware and Embedded Systems-CHES 2008. Berlin: Springer, 2008: 181–197

[17] Maes R, van Herrewege A, Verbauwhede I. PUFKY: a fully functional PUF-based cryptographic key generator // Workshop on Cryptographic Hardware and Embedded Systems-CHES 2012. Berlin: Springer, 2012: 302–319

[18] Maes R, Tuyls P, Verbauwhede I. Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs // Workshop on Cryptographic Hardware and Embedded Systems-CHES 2009. Berlin: Springer, 2009: 332–347

[19] Maes R, Tuyls P, Verbauwhede I. Soft decision helper data algorithm for SRAM PUFs // IEEE International Symposium on Information Theory-ISIT 2009. New York: IEEE, 2009: 2101–2105

[20] Schnabl G, Bossert M. Soft-decision decoding of Reed-Muller codes as generalized multiple concatenated codes. IEEE Transactions on Information Theory, 1995, 41(1): 304–308

[21] Maes R. An accurate probabilistic reliability model for silicon PUFs // Workshop on Cryptographic Hardware and Embedded Systems-CHES 2013. Berlin: Springer, 2013: 73–89