用于网络物理系统中连续消息认证的轻量级签名方案

LiS: Lightweight Signature Schemes for Continuous Message Authentication in Cyber-Physical Systems

---

基于国密算法的 PKI 在工控系统中的应用研究

报告人：韩守旭

本文涉及的相关专有名词和缩略词

- CPS(Cyber-physical system) 网络实体系统
- MAC(Message authentication code) 消息认证码
- CHF(Chameleon-hash functions)
- UHF(Universal hashing function)
- CPA(chosen-plaintext attack) 选择明文攻击
- PPT(probabilistic polynomial time) 概率多项式时间
- Computational complexity theory
- Probabilistic Turing machine
- BF(Bloom filter) 布隆过滤器

PKI 在工控系统中的应用

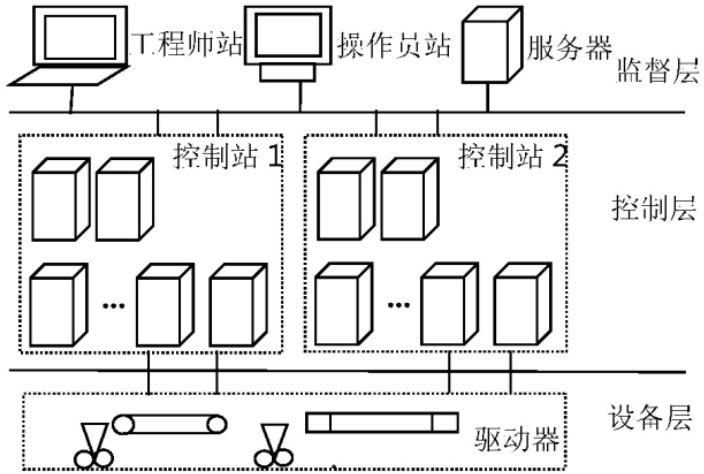


图 1　以 PLC 为中心的工控系统运行架构

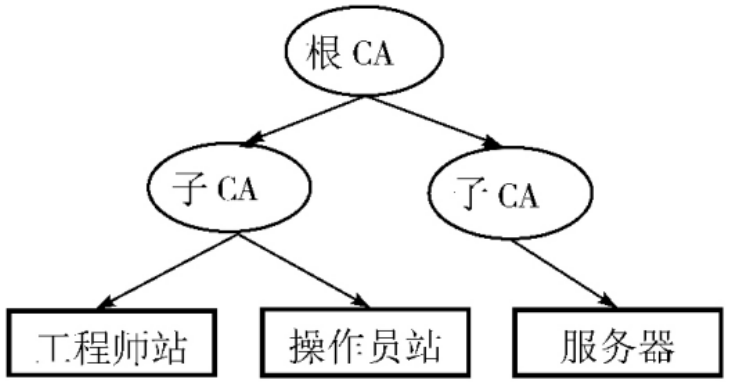


图 2　工控系统的证书认证模型

设置一个单一的严格信任根 CA 作为信任锚，2 个或多个子 CA 作为中间 CA，代替根 CA 对终端实体进行认证。终端实体一般是工控系统中的工程师站、操作员站、服务器等设备。认证的过程是根 CA 使用根私钥为子 CA 签发数字证书，子 CA 为终端实体签发数字证书。 有新的终端实体要加入可信任的认证链中时，其持有的证书要能够完成到根 CA 的层级认证

魏珊珊, 韩庆敏, 郭肖旺, 等. 基于国密算法的 PKI 在工控系统中的应用研究[J]. 计算机与现代化, 2018 (11): 1-6.

PKI 在工控系统中的应用

①新终端用户向 PKI 管理系统提交数字证书申请

②CA 审核用户的证书申请，并为用户生成公私钥对并签发用户证书，一并放入 USB KEY 中交付给用户，用户具有该 UK 唯一的口令

③用户使用自己的 USB KEY 设备，申请加入工 控系统的监督层。监督层有专门的设备审核用户的 数字证书

④监督层根据证书验证策略验证证书的真伪，验 证通过后，将允许新终端用户加入到监督层

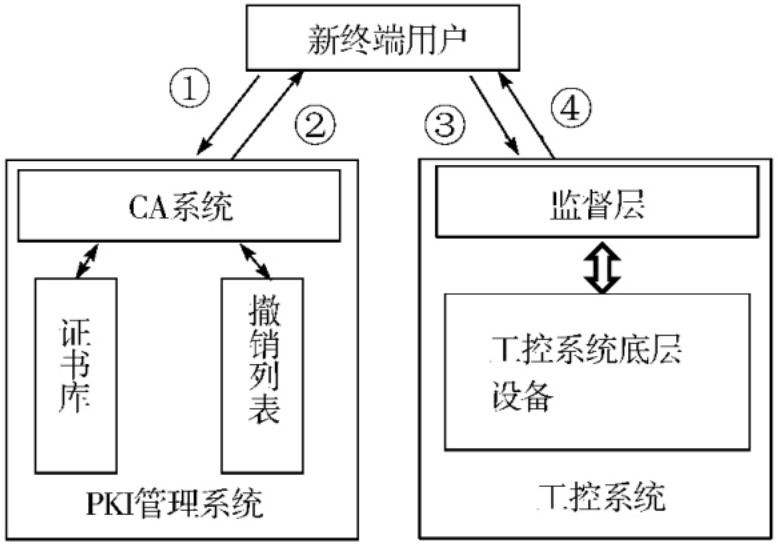


图 3　PKI 管理系统应用到工控系统

终端用户、PKI 管理系统、工控系统 3 者之间的数据流向和交互过程

国密算法

- 公开的SM2、SM3、SM4三类算法，分别是非对称算法、哈希算法和对称算法，密钥长度和分组长度均为128位
- SM2椭圆曲线公钥密码算法是我国自主设计的公钥密码算法，包括SM2-1椭圆曲线数字签名算法，SM2-2椭圆曲线密钥交换协议，SM2-3椭圆曲线公钥加密算法，分别用于实现数字签名密钥协商和数据加密等功能
- 并没有给出推荐的曲线，国密局推荐使用素数域256 位椭圆曲线，其曲线方程为 y^2= x^3+ax+b
- SM3算法：SM3杂凑算法是我国自主设计的密码杂凑算法，适用于商用密码应用中的数字签名和验证消息认证码的生成与验证以及随机数的生成，SM3算法的输出长度为256比特，因此SM3算法的安全性要高于MD5算法和SHA-1算法
- SM4分组密码算法是我国自主设计的分组对称密码算法，用于实现数据的加密/解密运算，以保证数据和信息的机密性，SM4算法与AES算法具有相同的密钥长度分组长度128比特

SM2 签名算法描述

SM2 签名算法描述: 首先用户输入原始明文数据，预处理之后对其进行哈希运算，哈希运算算法采用 SM3 算法。然后提取对应的私钥对哈希结果进行数字签名，得到签名结果文件输出到磁盘，等待下一步的签名验证。签名函数算法设计流程图见右图

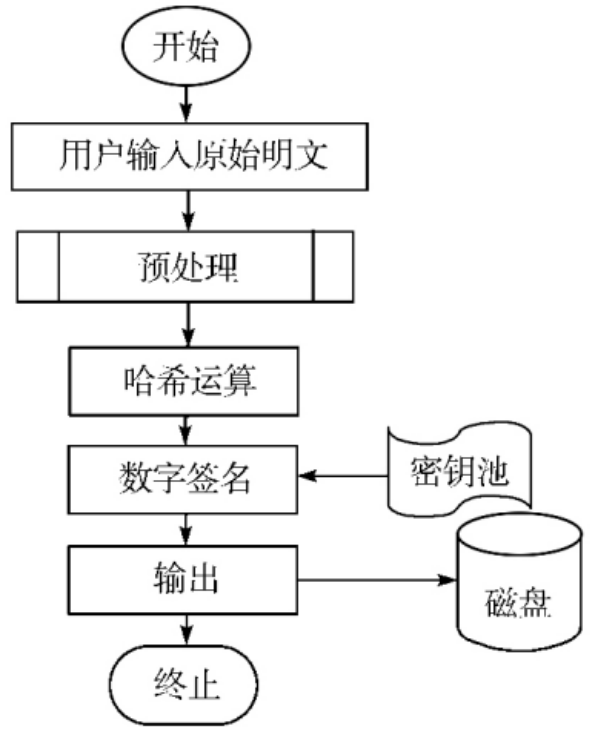


图 4　SM2 签名算法

SM2 验签算法描述

SM2 验签算法描述: 首先按照上文设计的 SM2 签名算法对原始明文重新计算签名，然后从证书池获取对应证书，提取公钥，对原始签名文件进行解密运算，对比签名结果和解密结果，若完全一致则证明验证签名通过，否则验证不通过。验签函数算法设计流程见右图

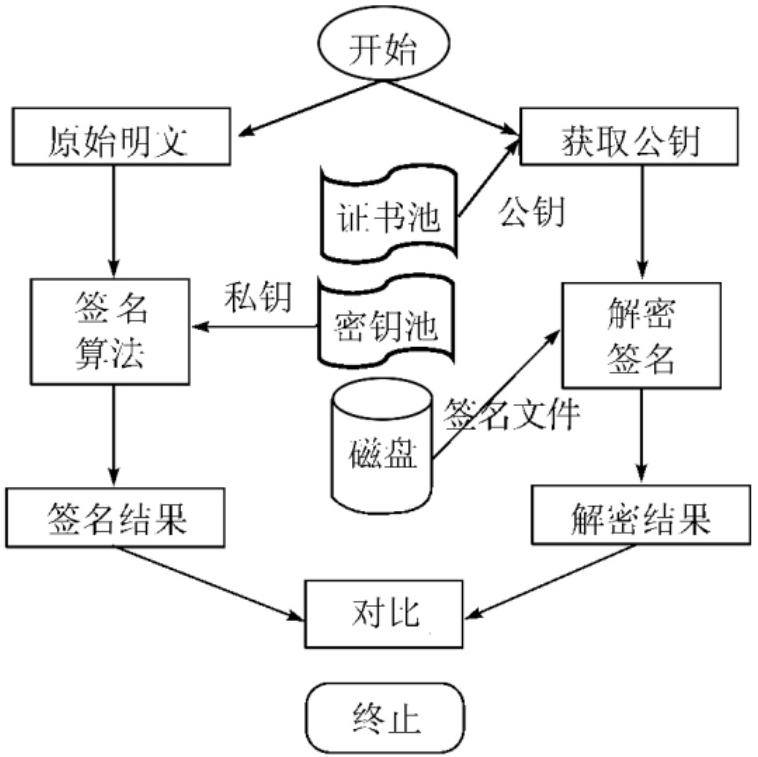


图 5　SM2 验签算法

用于网络物理系统中连续消息认证的轻量级签名方案

相关概念和密码原语

- 用 κ 表示安全参数
- $\varnothing$ 表示一个空字符串
- 用 $[n] = \{1, \ldots, n\} \subset N$ 表示介于 1 和 n 之间的整数集
- 如果 X 是一个集合，那么 $x \xleftarrow{\$} X$ 表示从 X 中采样一个均匀随机元素
- 如果 X 是一个概率算法，那么 $x \xleftarrow{\$} X$ 表示 X 使用新的 random coins 运行并返回 x
- 令 // 是连接两个字符串的操作
- $|\cdot|$ 是获取变量位长的操作
- # 是获取集合中元素数量的操作

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

UHF (Universal Hash Functions)

- 通用散列函数（UH）族：$K_{UH} \times M_{UH} \rightarrow R_{UH}$　　K_{UH}、M_{UH} 和 R_{UH} 是密钥、消息和输出空间

- 定义 2.1：我们说一组散列函数 UH 是 Universal Hash Functions，其满足：

  1）我们通过对随机散列密钥 $k \xleftarrow{\$} K_{UH}$ 进行采样来统一选择一个散列函数 $UHF \in UH$

  2）任意$(x, y) \in M_{UH}$ 我们有概率 $Pr[UHF(k, x) = UHF(k, y)] \leqslant 1/\#M_{UH}$

用于网络物理系统中连续消息认证的轻量级签名方案

CHF (Chameleon-hash functions)

- 也称为 trapdoor-hash functions，是一种具有陷门功能的哈希函数，它允许人们在函数域中找到任意碰撞。 然而，只要不知道相应的陷门（或密钥），Chameleon-hash functions 就是抗碰撞的。更准确地说，知道陷门的一方能够在函数域中找到相应碰撞

- 给定消息和随机性对 (m', r')，CHF 允许使用其秘密密钥 sk 有效地计算消息 $m \neq m'$ 的碰撞 r，使得 CHF(m, r) = CHF(m', r')

Chameleon-Hashes with Ephemeral Trapdoors And Applications to Invisible Sanitizable Signatures（https://eprint.iacr.org/2017/011.pdf）
Chameleon Signatures（https://www.ndss-symposium.org/ndss2000/chameleon-signatures/）

用于网络物理系统中连续消息认证的轻量级签名方案

CHF (Chameleon-hash functions)

- $CH(pk, \cdot, \cdot) : PK_{CH} \times M_{CH} \times R_{CH} \rightarrow Y_{CH}$ 与一对公钥 $pk \in PK_{CH}$ 和私钥 $sk \in SK_{CH}$ 相关联
- (PK_{CH}, SK_{CH}) 是公钥和私钥空间， M_{CH} 是消息空间，R_{CH} 是随机空间，Y_{CH} 是输出空间
- 这些公私钥对由 PPT 算法 $(pk, sk) \xleftarrow{\$} CHKGen(1^{\kappa})$ 生成
- 如果 key 在上下文中清楚描述，我们将为 CH(pk, m, r) 写 CH(m, r).

Chameleon-Hashes with Ephemeral Trapdoors And Applications to Invisible Sanitizable Signatures（https://eprint.iacr.org/2017/011.pdf）
Chameleon Signatures（https://www.ndss-symposium.org/ndss2000/chameleon-signatures/）

用于网络物理系统中连续消息认证的轻量级签名方案

CHF(Chameleon-hash functions)

- CH(m, r) 在输入消息 m 和随机字符串 r 时生成的哈希值满足以下属性:
  - 抗碰撞。 没有有效的算法可以在输入公钥 pk 时输出两对 (m1,r1) 和 (m2,r2)，使得 $m1 \neq m2$ 且 CH(m1,r1) = CH(m2,r2)
  - 陷门碰撞。 存在一个有效的确定性算法 CHColl，在输入密钥 sk 和 $(r, m, m') \in R_{CH} \times M_{CH} \times M_{CH}$ 时，输出值 $r' \in R_{CH}$ 使得 CH(pk,m,r) = CH(pk,m',r')
  - 均匀性。 对于CHKGen 输出的任意公钥pk，所有消息$m \in M_{CH}$ 在随机均匀绘制$r \xleftarrow{\$} R_{CH}$ 时生成均等分布的哈希值CH(m,r)。 此属性确保第三方无法通过推断有关散列消息的任何信息来检查值散列

Chameleon-Hashes with Ephemeral Trapdoors And Applications to Invisible Sanitizable Signatures（https://eprint.iacr.org/2017/011.pdf）
Chameleon Signatures（https://www.ndss-symposium.org/ndss2000/chameleon-signatures/）

用于网络物理系统中连续消息认证的轻量级签名方案

CHF(Chameleon-hash functions)

给定一个对手 A 和一个变色龙函数 CHF，CH 安全博弈被定义为：$G^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa)$

Proc.Init() :	**Proc.Finalize**(m, m', r, r') :
$(sk, pk) \xleftarrow{\$} \mathrm{CHKGen}(1^\kappa)$	If $(m, m') \in \mathcal{M}_{\mathrm{CH}} \wedge$
OUTPUT pk	$(r, r') \in \mathcal{R}_{\mathrm{CH}} \wedge m \neq m' \wedge$
	$\mathrm{CHF}(m, r) = \mathrm{CHF}(m', r')$
	OUTPUT 1
	ELSE OUTPUT 0

Figure 1: Procedures used to define security for CH.

Chameleon-Hashes with Ephemeral Trapdoors And Applications to Invisible Sanitizable Signatures（https://eprint.iacr.org/2017/011.pdf）
Chameleon Signatures（https://www.ndss-symposium.org/ndss2000/chameleon-signatures/）

用于网络物理系统中连续消息认证的轻量级签名方案

CHF(Chameleon-hash functions)

定义2.2：

- 用 $\mathrm{Adv}^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa) := \Pr[G^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa) = 1]$ 表示对手A在安全参数κ下破坏 Chameleon-hash functions 的安全性的优势
- 如果 PPT 对手没有具有不可忽略的优势 $\mathrm{Adv}^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa)$ 我们就说 CHF 是安全的

Chameleon-Hashes with Ephemeral Trapdoors And Applications to Invisible Sanitizable Signatures（https://eprint.iacr.org/2017/011.pdf）
Chameleon Signatures（https://www.ndss-symposium.org/ndss2000/chameleon-signatures/）

Digital Signature Schemes

- 定义了一个数字签名方案 SIG，签名方案与安全参数 κ 中的公钥和私钥空间{PK_{SIG}, SK_{SIG}}、消息空间 M_{SIG} 和签名空间S_{SIG} 相关联。 我们用 ℓ_s 表示空间 R_s 的位长，它由 κ 决定。 SIG的算法定义如下:
  - KGen(1^k, ℓ, aux)：该算法以安全参数1^k、SIG可以生成的最大签名数 ℓ 和辅助输入 aux 为输入，生成密钥 sk 和验证密钥 vk
  - Sign(sk, m)：这是一种签名算法，它使用签名密钥 sk 为消息m ∈ M_{SIG} 生成签名σ ∈ S_{SIG}
  - Verify(vk, m, σ)：这是一种验证算法，它以验证密钥 vk、消息 m 和签名 σ 作为输入，如果σ 是 vk 下 m 的有效签名，则输出1，否则输出0

用于网络物理系统中连续消息认证的轻量级签名方案

- 安全博弈: $G^{\mathrm{SIG}}_{\mathcal{A},\mathrm{S}}(\kappa, \ell)$
- 对手 A 通过依次调用过程 Proc.Init、Proc.SQuery 和 Proc.Finalize 来进行

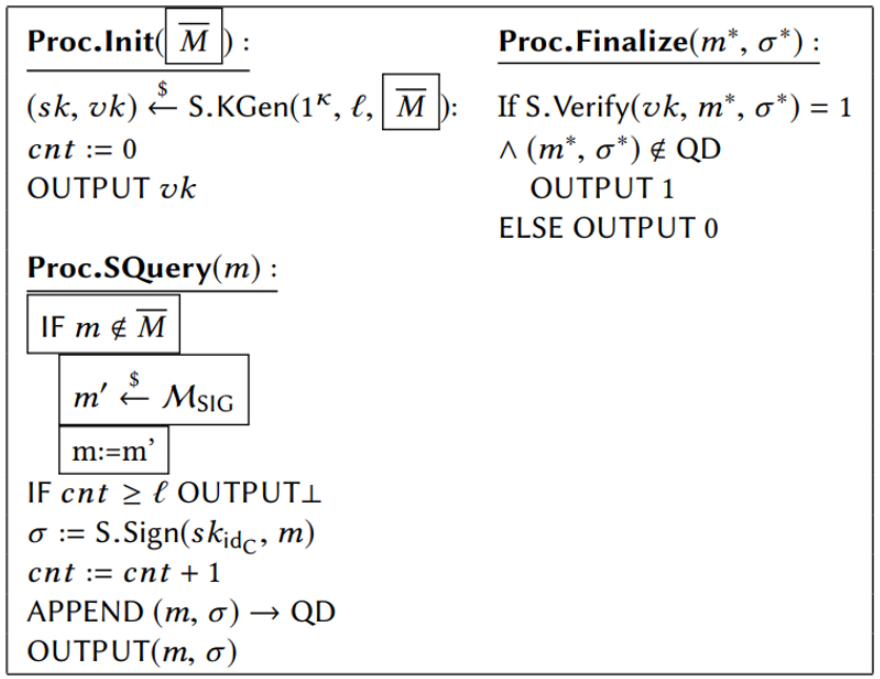


Figure 2: Procedures used to define security for SIG.

定义2.3：

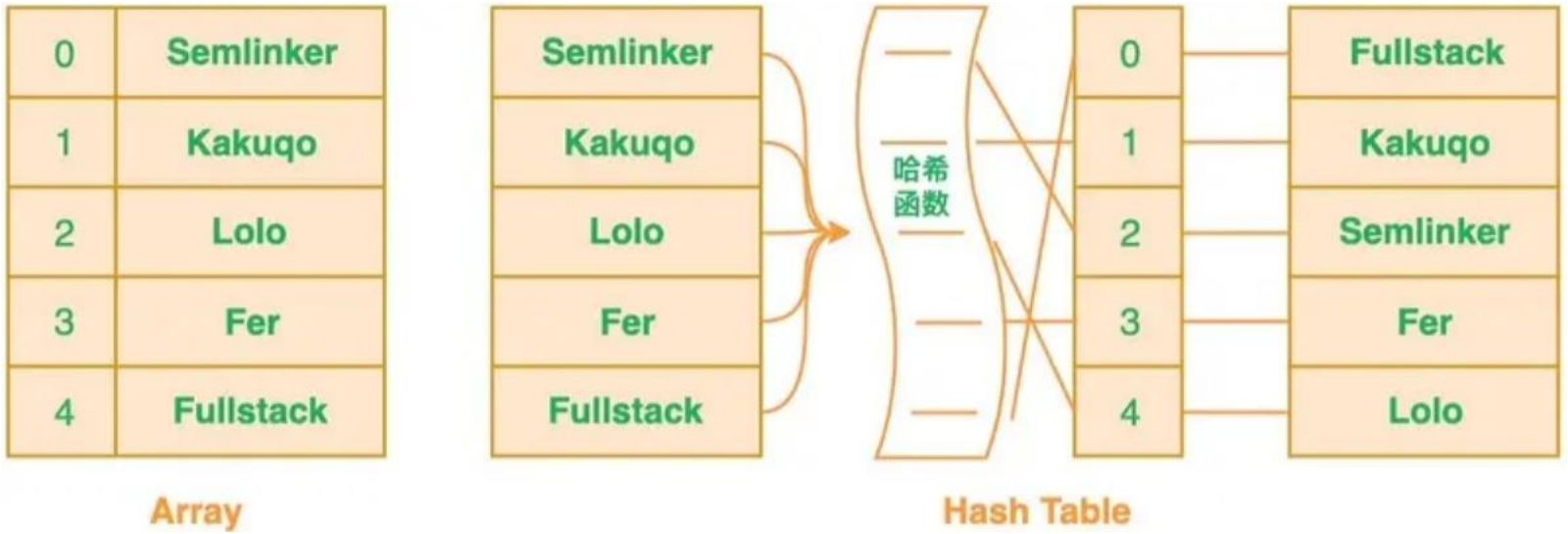
- $\mathrm{Adv}^{\mathrm{SIG}}_{\mathcal{A},\mathrm{S}}(\kappa,\ell) := \left|\Pr[G^{\mathrm{SIG}}_{\mathcal{A},\mathrm{S}}(\kappa,\ell)=1]-\frac{1}{2}\right|$ 为 PPT 对手 A 在安全参数 κ 下破坏 SIG 自然方案 S 的安全性优势
- 如果 PPT 对手没有具有不可忽略的优势：$\mathrm{Adv}^{\mathrm{SIG}}_{\mathcal{A},\mathrm{S}}(\kappa,\ell)$. 我们说 S 是安全的

Bloom filter（布隆过滤器）

- 布隆过滤器是一种概率数据结构，它提供了一个集合的空间高效存储，并且可以有效地测试一个元素是否是该集合的成员

  - Init(N, ϵ)：在输入时，设置大小为 N，初始化算法启动位长为 $1.44\epsilon N$ 的布隆过滤器
  - Insert(m)：元素插入算法以元素m为输入，将m插入到BF中。
  - Check(m)：如果元素m 在BF 中，则元素检查算法返回1，否则返回0
  - Pos(m)：位置更新算法计算BF 中元素m 要更改的位置

Bloom filter

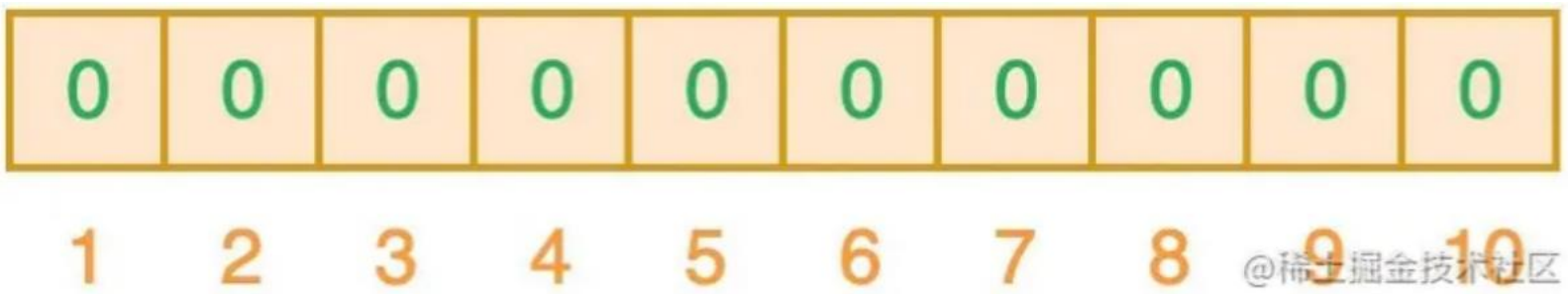


- 当你往简单数组或列表中插入新数据时，将不会根据插入项的值来确定该插入项的索引值
- 利用哈希表你可以通过对 “值” 进行哈希处理来获得该值对应的键或索引值，然后把该值存放到列表中对应的索引位置

https://zhuanlan.zhihu.com/p/43263751 & https://juejin.cn/post/6844904007790673933 & https://en.wikipedia.org/wiki/Bloom_filter

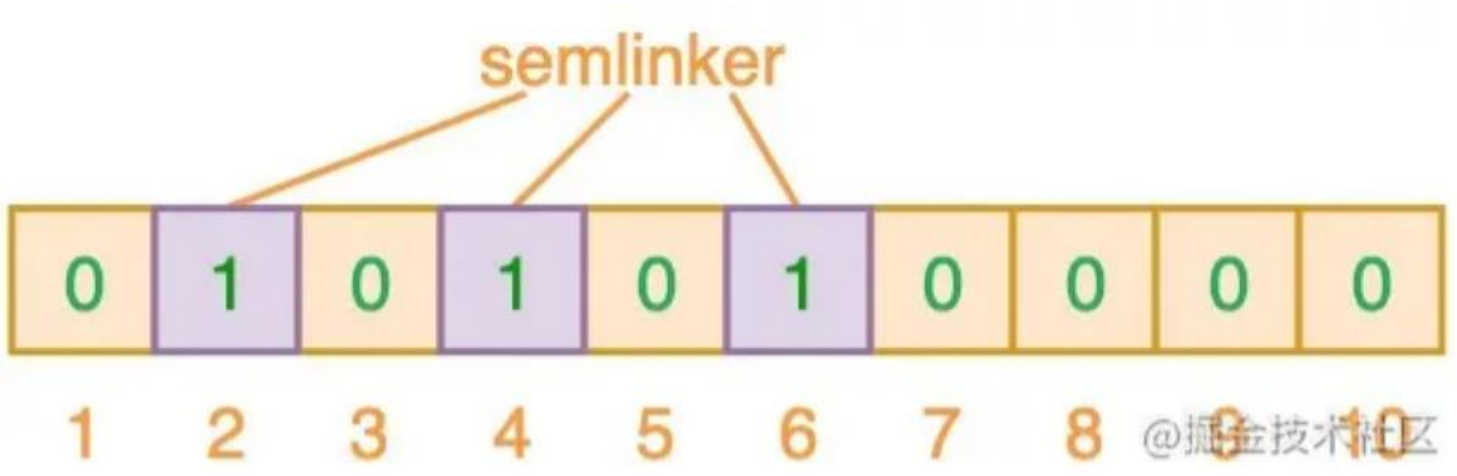
Bloom filter

- 布隆过滤器可以检查值是 **“可能在集合中”** 还是 **“绝对不在集合中”**。“可能” 表示有一定的概率，也就是说可能存在一定为误判率
- 布隆过滤器（Bloom Filter）本质上是由长度为 m 的位向量或位列表（仅包含 0 或 1 位值的列表）组成，最初所有的值均设置为 0



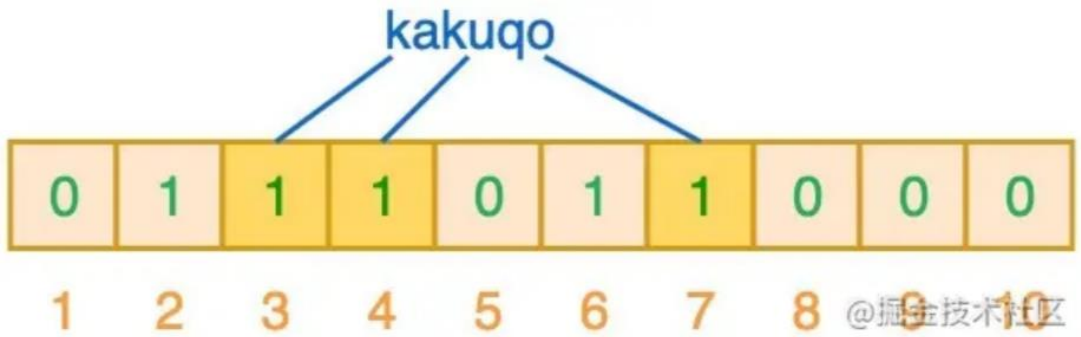
Bloom filter

- 为了将数据项添加到布隆过滤器中，我们会提供 K 个不同的哈希函数，并将结果位置上对应位的值置为 “1”

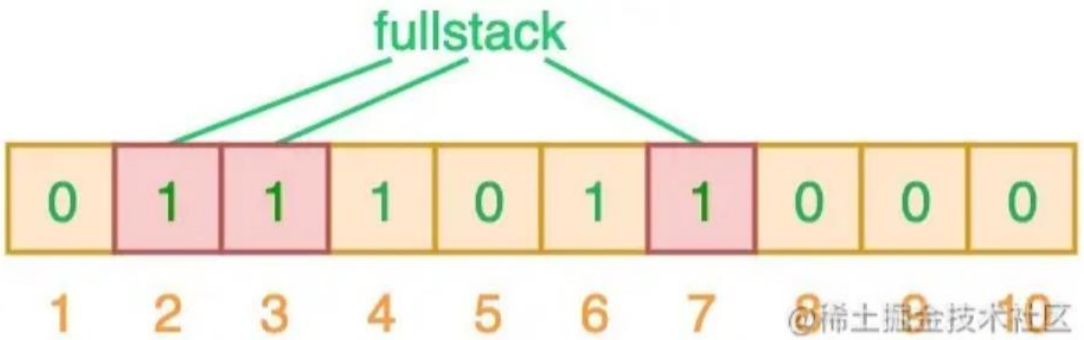


- 在前面所提到的哈希表中，我们使用的是单个哈希函数，因此只能输出单个索引值。而对于布隆过滤器来说，我们将使用多个哈希函数，这将会产生多个索引值

Bloom filter



- 另一个输入 “kakuqo”，哈希函数输出 3、4 和 7,索引位 4 已经被先前的 “semlinker” 标记



- 相应的索引位都被置为 1，这意味着我们可以说 “fullstack” 可能已经插入到集合中。事实上这是误报的情形，是由于哈希碰撞导致的巧合而将不同的元素存储在相同的比特位上

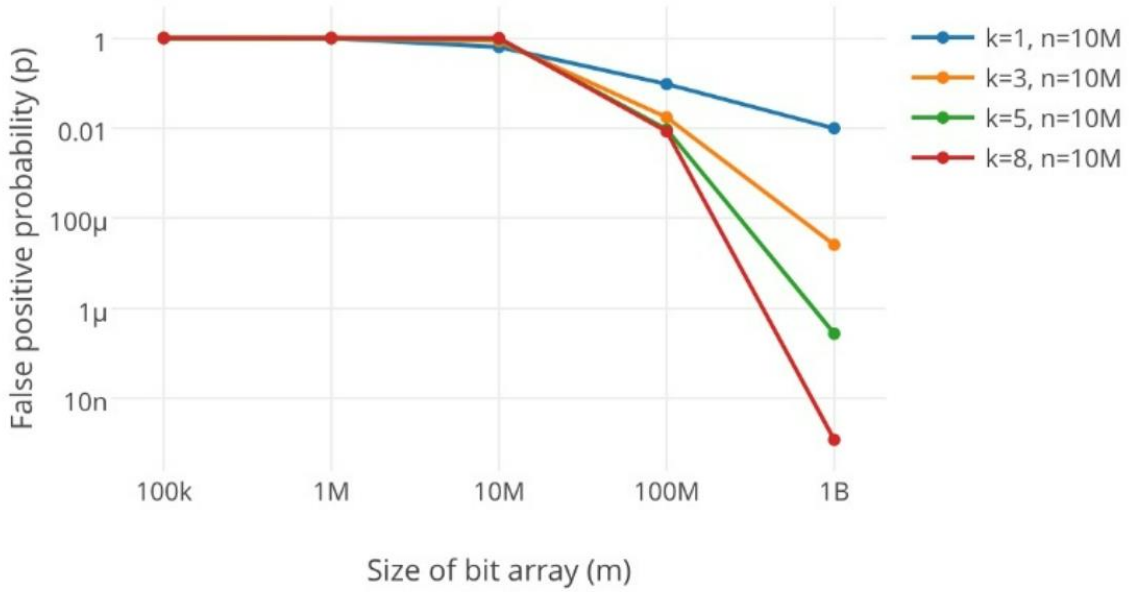
哈希函数个数和布隆过滤器长度

- 选择适合的 k 和 m 值：

$$m = -\frac{n \ln p}{(\ln 2)^2} \qquad k = \frac{m}{n} \ln 2$$

- 相应的误判率：

$$\left(1 - e^{-kn/m}\right)^k$$



k 为哈希函数个数，m 为布隆过滤器长度，n 为插入的元素个数，p 为误报率

用于网络物理系统中连续消息认证的轻量级签名方案

- 利用 Chameleon-hash functions 和预计算策略，将计算和存储的“负担”从签名者转移到验证者以及可能受信任的第三方服务器， CPS 中的验证者是具有足够计算能力和存储的服务器
- 利用受信任的服务器将一组虚拟消息/随机性对 (m'_i,r'_i)（$1 \leqslant i \leqslant \ell$）作为输入，并预先计算将使用的变色龙哈希值作为验证密钥 vk 的一部分
- 对于在线签名生成，签名者只需要根据使用的虚拟随机 r'_i 计算碰撞 r_i 作为 m_i 的签名

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- 为了进一步优化资源受限签名者的签名算法，我们建议将所有虚拟消息 m'_i 固定为常数 M
- 并使用通用哈希函数 UHF 链接所有虚拟随机性，即 $r'_i = UHF(k, r'_{i-1})$ for $1 \leqslant i \leqslant \ell$，并且 r_0 是随机选择的，其中 k 是 UHF 的随机散列密钥
- 签名者只需要存储几百个比特（$sk, sk \cdot M, r'_0, k$）即可进行签名
- LiS1 中只需要三个模加法和两个模乘法来生成一个签名

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- 在上述朴素的构造中，验证密钥的大小由预先计算的哈希值决定。 如果 vk 可以支持多个签名的验证（即 ℓ 很大），那么 vk 也可能变得非常大。 为了减小 vk 的大小，我们建议使用布隆过滤器来压缩 vk
- 但验证密钥仍然会在ℓ签名后用完
- 进一步开发了两种验证密钥补充解决方案，以实现无限制的签名能力

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

LIS: LIGHTWEIGHT SIGNATURE SCHEMES FROM CHAMELEON HASH

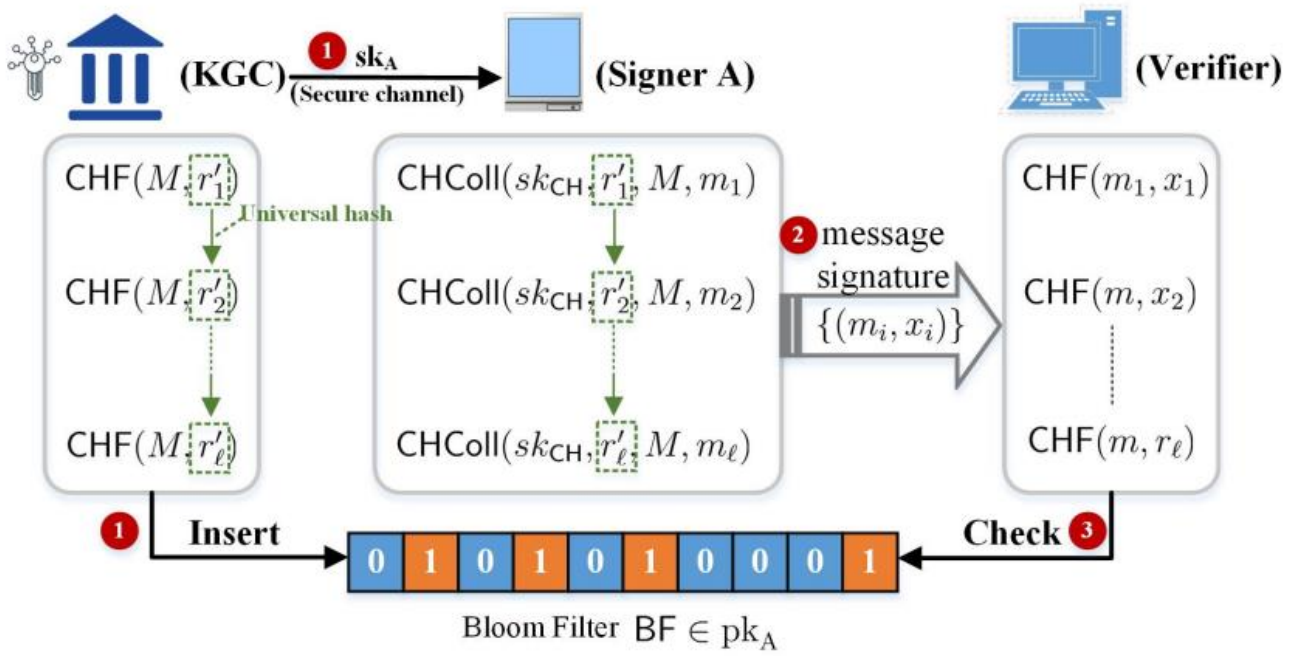


Figure 3: Overview of LiS. KGC stands for *key generation center*.

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS1:**

  该方案依赖于通用哈希函数 UHF、 Chameleon-hash functions CHF 和布隆过滤器 BF。所提出方案 LiS1 的算法如图 4 所示:

$\mathbf{LiS}_1.\mathbf{KGen}(1^\kappa, \ell, \text{aux})$:

$(sk_{CH}, pk_{CH}) \xleftarrow{\$} \text{CHKGen}(1^\kappa)$
$k \xleftarrow{\$} \mathcal{K}_{UH}$; $\epsilon \leftarrow \text{aux}$
$\text{BF.Init}(\ell, \epsilon)$
$M \xleftarrow{\$} \mathcal{M}_{CH}$; $r'_0 \xleftarrow{\$} \mathcal{R}_{UH}$
For $i \in [\ell]$:
  $r'_i := \text{UHF}(k, r'_{i-1})$
  $t_i := \text{CHF}(M, r'_i)$
  $\text{BF.Insert}(t_i)$
$r' := r'_1$; $sk_{\text{id}_C} := (sk_{CH}, k, r', M)$
$vk_{\text{id}_C} := (\text{BF}, pk_{CH})$
Return $(sk_{\text{id}_C}, vk_{\text{id}_C})$

$\mathbf{LiS}_1.\mathbf{Sign}(sk_{\text{id}_C}, m)$:

$x := \text{CHColl}(sk_{CH}, r', M, m)$
$r' := \text{UHF}(k, r')$
Return x

$\mathbf{LiS}_1.\mathbf{Verify}(vk_{\text{id}_C}, m, x)$:

$vr := 0$
$t := \text{CHF}(m, x)$
$vr := \text{BF.Check}(t)$
Return vr

Figure 4: Algorithms of LiS_1.

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS1过程 -- 初始化:**

- 签名者 id_C 首先运行 Chameleon-hash functions $(sk_{CH},pk_{CH}) \xleftarrow{\$} CHKGen(1^k)$ 的密钥生成算法生成一对秘密/公钥
- 采样一个随机密钥 $k \xleftarrow{\$} K_{UH}$ 用于通用哈希函数 UHF
- 采样随机消息 $M \xleftarrow{\$} M_{CH}$，以及初始随机值 r'_0
- 通过参数 aux，id_C 解析出布隆过滤器的“假阳性参数”ϵ
- 布隆过滤器实例 BF 由 $BF.Init(\ell, \epsilon)$ 初始化。
- 对于 $i \in [\ell]$，id_C 生成 ℓ 虚拟随机值，使得 $r'_i := UHF(k, r'_{i-1})$，以及验证点 $t_i := CHF(M, r'_i)$
- id_C 将这些验证点插入到布隆过滤器 $BF.Insert(t_i)$ 中。随机变量 $r' := r'_1$ 用于生成下一个签名。
- 请注意，如果任何 $r'_i = r'_j$ for $i \neq j$ 则 id_C 重新运行密钥生成算法。
- 最终，id_C 的密钥和验证密钥是 $sk_{id_C} := (sk_{CH}, k, r', M)$ 和 $vk_{id_C} := (BF, pk_{CH})$
- 验证密钥 $vk_{id_C} := (BF,pk_{CH})$ 将被发送给潜在的验证者，并且密钥 sk_{id_C} 将由签名者 id_C 私人存储
- 为了验证第一个验证密钥，签名者可以通过安全通道（与对手隔离）将其传输到指定的验证者，或者要求可信赖的第三方对其进行数字签名

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS1过程 -- 签名:**

  - 在获得需要认证的消息 m 后，id_C 首先检索存储的密钥 $sk_{id_C} := (sk_{CH}, k, r', M)$。
  - 由于 CHF 的陷门碰撞属性，id_C 可以将 m 的签名 x 计算为 $x := CHColl(sk_{CH}, r', M, m)$。
  - 然后，id_C 可以将 m 与签名 x 一起发送给验证者
  - 在此之后，id_C 将虚拟随机性 r' 更新为下一个，如 $r' := UHF(k, r')$。
  - 事实上，在进行下一次消息认证之前，可以随时更新虚拟随机性，因此它的性能开销可能隐藏在后台

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS1过程 --验证:**

  - 验证者 id_S 接受收到来自签名者 id_C 的消息 m 及其签名 x
  - 验证者 id_S 通过检查结果哈希值 t = CHF(m, x) 是否在布隆过滤器中来验证它，即 BF.Check(t)

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- 由于布隆过滤器没有任何假阴性，对于每个 $x := LiS1.Sign(sk_{id_C}, m)$，它必须有 $BF.Check(CHF(m, x)) = 1$ 因为 $CHF(m, x) = CHF(m'_i, t_i)$ 在初始化期间插入到 BF 中

- 为了获得更好的在线效率，签名者可以离线（或在其空闲时间）计算通用哈希操作。 当然，签名者也可以预先计算和缓存许多这样的通用哈希值。 然后签名者只需要在在线签名阶段运行 CHColl，因此签名算法可以快 2 倍左右

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS2:**

$\mathbf{LiS_1.KGen}(1^\kappa, \ell, \text{aux})$:

$(sk_{CH}, pk_{CH}) \xleftarrow{\$} \text{CHKGen}(1^\kappa)$
$k \xleftarrow{\$} \mathcal{R}_{CH}; \epsilon \leftarrow \text{aux}$
$\text{BF.Init}(\ell, \epsilon)$
$M \xleftarrow{\$} \mathcal{M}_{CH}$
For $i \in [\ell]$:
  $r'_i := h_2(k||i)$
  $t_i := \text{CHF}(M, r'_i)$
  $\text{BF.Insert}(t_i)$
$cnt := 1;$
$sk_{id_C} := (sk_{CH}, k, M)$
$vk_{id_C} := (\text{BF}, pk_{CH})$
Return (sk_{id_C}, vk_{id_C})

$\mathbf{LiS_2.Sign}(sk_{id_C}, m)$:

$N \xleftarrow{\$} \mathcal{R}_2$
$y := h_1(m||N)$
$x := \text{CHColl}(sk_{CH}, r'_{cnt}, M, y)$
$cnt := cnt + 1$
$r'_{cnt} := h_2(k||cnt)$
Return x, N

$\mathbf{LiS_2.Verify}(vk_{id_C}, (m, N), x)$:

$vr := 0$
$t := \text{CHF}(h_1(m||N), x)$
$vr := \text{BF.Check}(t)$
Return vr

Figure 5: Algorithms of LiS_2.

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS2过程 -- 初始化:**
  - 签名者id_C首先运行变色龙哈希函数$(sk_{CH},pk_{CH}) \xleftarrow{\$} CHKGen(1^{\kappa})$的密钥生成算法，生成一对秘密/公钥
  - 采样一个随机密钥$k \xleftarrow{\$} R_{CH}$，一条随机消息 $M \xleftarrow{\$} M_{CH}$
  - id_C 将另外初始化两个加密哈希函数 $h1: \{0,1\}^* \rightarrow M_{CH}$ 和 $h2: \{0,1\}^* \rightarrow R_{CH}$
  - 布隆过滤器实例 BF 由 $BF.Init(\ell,\epsilon)$ 初始化
  - 对于 $i \in [\ell]$，id_C 生成 ℓ 虚拟随机值，使得 $r'_I := h2(k||i)$，以及验证点 $t_i := CHF(M, r'_i)$ 以备将来使用
  - 同时，id_C 将这些验证点插入到 Bloom 过滤器 $BF.Insert(t_i).id_C$ 初始化一个计数器 cnt = 0 来计算生成签名的数量
  - 作为该过程的最终结果，id_C 的秘密密钥和公共验证密钥是 $sk_{idC}:=(sk_{CH}, k, M, cnt)$ 和 $vk_{idC}:=(BF, pk_{CH})$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS2过程 -- 签名:**

  - 为了验证消息 m，id_C 首先采样一个随机值 $N \xleftarrow{\$} R_2$
  - 计算 $y := h(m||N)$ 和 $r'_{cnt} := h(k||cnt)$
  - 更新计数器 $cnt = cnt + 1$
  - 然后签名者 id_C 为 y 生成签名 x 为 $x := CHColl(sk_{CH}, r'_i, M, y)$，并发送元组 (m, N, x) 给验证者

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **LiS2过程 -- 验证:**

  - 接收到 (m, N, x) 后
  - 验证者 id_S 通过检查结果哈希值 $t = CHF(h_1(m||N), x)$ 是否在布隆过滤器中来验证它

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

LiS1 安全性分析

定理 4.1 我们假设变色龙散列函数 CHF 和通用散列函数 UHF 是安全的

那么具有给定参数 κ 和 ℓ 的 LiS_1 是安全的，可以抵御选择性选择的消息攻击，在具有优势：

$$\mathrm{Adv}^{\mathrm{SIG}}_{\mathcal{A},\mathrm{LiS}_1}(\kappa,\ell) \leq \mathrm{Adv}^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa) + 2^{-\epsilon}$$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

4.1 定理证明

- 让 BK_i 表示在游戏 i 中存在对手 A 获胜的事件。 BK_0是A在真实游戏中攻破方案的优势。下面，我们将改变从真实游戏开始的游戏，直到最后一场 A 的优势为零的游戏

- **游戏0**。这个游戏相当于数字签名方案的真实 SEUF-wCMA 安全实验。 同时，根据LiS_1的规范，诚实地回答所有查询。 因此，我们有

$$\Pr[\mathrm{BK}_0] = \mathrm{Adv}_0 = \mathrm{Adv}^{\mathrm{SIG}}_{\mathcal{A},\mathrm{LiS}_1}(\kappa, \ell).$$

4.1 定理证明

- **游戏 1**. 这个游戏像以前一样进行，但挑战者使用 ℓ 均匀分布的虚拟随机性 $\{r'_i\}_{i \in [\ell]}$ 来生成验证密钥，而不使用通用哈希函数 UHF。 这种修改实际上并没有改变虚拟随机性的分布。 回想一下，UHF 的种子是一个随机值 r_0，在我们的设计中 UHF 的哈希键也是随机的。 因此，根据定义 2.1，UHF 的输出是均匀分布的。 因此，我们可以随机选择那些虚拟随机性 $\{r'_i\}_{i \in [\ell]}$。 因此，我们有

$$\Pr[\mathrm{BK}_1] = \Pr[\mathrm{BK}_0].$$

用于网络物理系统中连续消息认证的轻量级签名方案

- **游戏 2**. 这个游戏像以前一样进行，但挑战者改变游戏如下。 它为每条消息 $m_i \in M$（由 A 提交）选择随机性 $x_i \$\leftarrow R_{CH}$ 作为签名，而不是运行碰撞生成函数 CHColl。 验证密钥是使用真实签名和消息对生成的，即 $\{(x_i,m_i)\}i\in[\ell]$。 注意 $x_i := M \cdot sk_{CH} + r'_i - m_i \cdot sk_{CH} \pmod q$。 我们可以将 x_i 重写为 $x_i:= r'_i + \tilde{m}_i \pmod q$ 其中 $\tilde{m}_i = M \cdot sk_{CH} - m_i \cdot sk_{CH}$。 由于每个 m_i 都是唯一的，因此 $\tilde{m}_i$ 也是唯一的。 我们声称每个 x_i 在统计上接近于距离为 0 的均匀随机值。这一说法的证明可以遵循 [49, 引理 1] 的组合函数的证明。 因此，我们有

$$\Pr[BK_2] = \Pr[BK_1].$$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **游戏 3**. 在这个游戏中，挑战者 C 与前一游戏完全一样，但增加了一个中止规则。也就是说，如果对手提交一个元组 (m*, x*) 导致与 BF 中记录的那些哈希值之一发生冲突，即 CHF(m*, x*) = CHF(M,r'$_i$) 对于某些 r'$_i$。如果这种情况以不可忽略的概率 $\mathrm{Adv}^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa)$ 发生，那么我们可以通过使用 A 来破坏变色龙哈希函数的安全性来构造一个有效的算法 F

- 具体来说，F 可以模拟 A 的签名游戏，同时从变色龙哈希挑战者那里接收挑战公钥 $\mathrm{pk}_{\mathrm{CH}}$。然而，在不知道密钥 $\mathrm{sk}_{\mathrm{CH}}$ 的情况下，F 无法像 LiS_1 那样在线计算冲突。相反，F 使用随机选择作为之前游戏的真实签名来生成验证密钥，而不是使用虚拟消息/随机性对。由于之前游戏中的修改，此更改是可能的。由于变色龙哈希函数的安全性，我们有：

$$\Pr[\mathrm{BK}_2] \leq \Pr[\mathrm{BK}_3] + \mathrm{Adv}^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa).$$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

- **游戏 4**. 在这个游戏中，C 像以前一样进行，但是如果对手 A 提交一个元组 (m^*, x^*) 使得 $\text{Check}(\text{CHF}(m^*, x^*)) = 1$ 和 (m^*, x^*) 之前没有被对手查询过，即 A 发现了 BF 的误报。 通过应用BF的误报概率，我们有

$$\Pr[\text{BK}_3] \leq \Pr[\text{BK}_4] + 2^{-\epsilon}.$$

- 在这个游戏中，如果 C 没有中止，那么 Proc.Finalize 查询将始终返回 0。因此，A 在这个游戏中的优势为零，即 $\Pr[\text{BK}_4] = 0$。将上述游戏中的概率放在一起， 我们有以下概率

$$\text{Adv}_0 \leq \text{Adv}^{\text{CH}}_{\mathcal{A},\text{CHF}}(\kappa) + 2^{-\epsilon}$$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

LiS1 安全性分析

- **GAME 0** : 遵循原始算法的真实实验
- **GAME 1** : 用统一的随机值替换通用哈希函数 UHF 的每个输出
- **GAME 2** : 为每条消息 m_i 随机生成一个签名 x_i 而不是运行 CHColl
- **GAME 3** : 将安全性降低到变色龙哈希函数 CHF 的安全性
- **GAME 4** : 降低对 Bloom Filter BF 误报错误的安全性

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

LiS2 安全性分析

定理 4.2。 我们假设变色龙散列函数 CHF 是安全的，并且散列函数 h 被建模为随机预言机。 然后，具有给定参数 κ 和 ℓ 的 LiS_2 可以安全地抵御自适应选择消息攻击，并具有优势:

$$\mathrm{Adv}^{\mathrm{SIG}}_{\mathcal{A},\mathrm{LiS}_2}(\kappa,\ell) \leq \mathrm{Adv}^{\mathrm{CH}}_{\mathcal{A},\mathrm{CHF}}(\kappa) + \frac{\ell^2}{2^{\ell_r}} + 2^{-\epsilon}$$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

实例化：

设 p 和 q 是两个大素数，使得 $p = u \cdot q + 1$ 其中 u 是一个小整数

我们特别有 $K_{CH} = M_{CH} = R_{CH} = K_{UH} = M_{UH} = S_{SIG} = Z_q$ 和 $Y_{CH} = Z_p$

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **哈希函数 h_1 和 h_2：**

  使用标准化的加密散列函数 SHA2：

  PUB FIPS. 2012. 180-4. Secure hash standard (SHS),âĂİ March (2012).

- **通用哈希函数 UHF：**

  Larry Carter and Mark N. Wegman. 1977. Universal Classes of Hash Functions (Extended Abstract). In STOC. ACM, 106–112

  通过以上提出的乘法模块化方案来实例化 UHF。 UHF 的密钥 $k = (k_0, k_1)$ 由两个群元素 $k_0 \$\leftarrow Z^*_q$ 和 $k_1 \$\leftarrow Z^*_q$ 组成。给定消息 m， 散列函数计算散列值 $y := UHF(k,m) = k_0 \cdot m + k_1 \pmod q$。 可以通过以下方式采用一些优化：

  Martin Dietzfelbinger, Torben Hagerup, Jyrki Katajainen, and Martti Penttonen. 1997. A Reliable Randomized Algorithm for the Closest-Pair Problem. J. Algorithms 25, 1 (1997), 19–51

用于网络物理系统中连续消息认证的轻量级签名方案

- **变色龙哈希函数的实例化：**

  Chameleon Signatures（https://www.ndss-symposium.org/ndss2000/chameleon-signatures/）

- **为了在碰撞算法中获得更好的性能，稍微修改了以上哈希评估算法：**

  - CHKGen(1^{κ})：密钥生成算法在 Z^*_p 中采样 q 阶随机组生成器 g 和一个秘密密钥 sk_{CH} \$← Z^*_q，并计算公钥 $pk_{CH} := g^{sk_{CH}} \pmod p$
  - CHF(pk_{CH}, m, r)：评估算法以公钥 $pk_{CH} \in Z^*_p$、消息 $m \in Z^*_q$ 和随机性 $r \in Z^*_q$ 作为输入，并输出哈希值 $y := g^r pk^m_{CH} \pmod p$。与原文中的算法相比，仅交换了m和r的位置，这种变化只是概念上的
  - CHColl(sk_{CH}, r', M, m)：一种高效的确定性碰撞算法 CHColl 将密钥 sk_{CH} 作为输入，并且 (r', M, m) $\in Z^*_q$，输出一个值 $x := M \cdot sk_{CH} + r' - m \cdot sk_{CH} \pmod q$
  - 签名者id_C可以预先计算$M \cdot sk_{CH}$并存储它(并非M)

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

- **验证密钥补充：**

预计算策略的一个限制是预计算的验证密钥最终会被用完，只刷新验证密钥而不修改私钥/公钥对（$\mathrm{pk}_{\mathrm{CH}}$，$\mathrm{sk}_{\mathrm{CH}}$）就足够了，因此签名者上运行的服务不会中断，设计为两种方案：

- Server-aided Replenishment (SAR): 服务器辅助补货
- Verifier Self-replenishment (VSR): 验证人自助补货

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

Server-aided Replenishment (SAR): 服务器辅助补货

- 签名者可以将新布隆过滤器实例 BF' 的重新初始化工作外包给受信任的服务器（不是验证者）
- 知道虚拟随机性/消息对 (r', pk^{M}_{CH}) 和通用哈希函数的密钥 k 的外包服务器可以为签名者计算那些变色龙哈希值，而无需任何交互
- 签名者不需要参与验证密钥的更新，并且可以继续使用其签名密钥对未来的消息进行连续签名
- 外包服务器（密钥生成中心）只需要定期发布一个新的 BF' 连同服务器的签名到一个公共公告，可供公众下载
- 签名者方面无需更改任何内容。 只要在旧的验证密钥失效之前及时补充验证密钥，签名者和验证者就可以并行运行

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

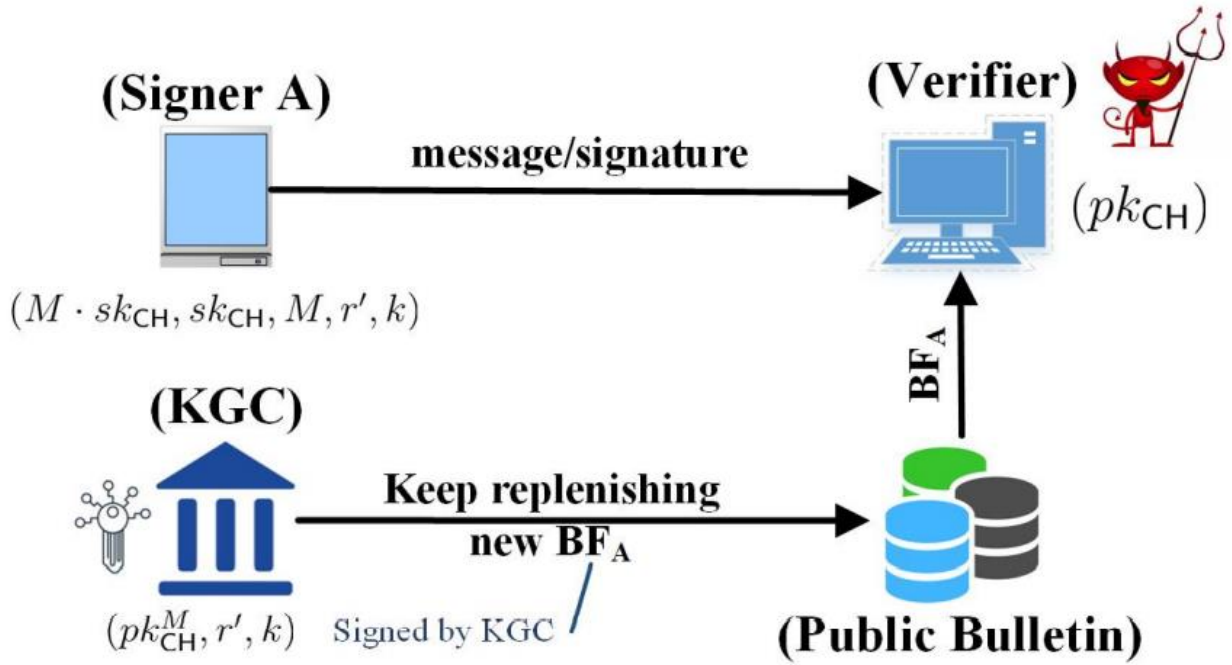


Figure 6: System Overview of Server-aided Replenishment (SAR). Verifiers can be malicious. The singer and the trusted key generation center (KGC) do not need any interactions during the replenishment procedure.

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

Verifier Self-replenishment (VSR): 验证人自助补货

- 如果验证者是值得信赖的（不受对手控制或破坏），那么我们可以允许验证者拥有 (pk^{M}_{CH}, r', k) 以进行签名验证。这样，验证者就可以定期补充自己的验证密钥
- 在信息物理系统中，消息在固定时间段内定期发送是很常见的。因此，我们实际上可以利用这一事实并开发一种简化的验证算法
- 在模型中，我们可以考虑所有消息都与一个单调递增的时间戳相关联,使用 T_m 表示消息 m 中的时间戳，并让 T_l 是收到最后一个有效签名的时间。 Δs 代表签名者发送的两个连续消息之间的固定时隙
- Setup 和 Verify 算法中都不再需要布隆过滤器

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

Verifier Self-replenishment (VSR): 验证人自助补货 --消息在固定时间段内定期发送

$\mathbf{LiS}_1^{\mathrm{VSR}}.\mathbf{Verify}(vk_{\mathrm{id_C}}, m, x)$:

$(k, pk_{\mathrm{CH}}^M, T_l) = vk_{\mathrm{id_C}}$
$r' \leftarrow$ *private storage*
If $T_m < T_l$: OUTPUT 0
$r'_0 := r'$; $vr := 0$; $t := \mathrm{CHF}(m, x)$
$\ell := \lfloor \frac{T_m - T_l}{\Delta_s} \rfloor$
For $i \in [\ell]$:
  $r'_i := \mathrm{UHF}(k, r'_{i-1})$
$t_\ell := pk_{\mathrm{CH}}^M \cdot g^{r'_\ell}$;
If $t_\ell = t$:
  $vr := 1$; $r' := r'_\ell$; $T_l := T_m$
OUTPUT vr

$\mathbf{LiS}_2^{\mathrm{VSR}}.\mathbf{Verify}(vk_{\mathrm{id_C}}, (m, N, cnt), x)$:

$(pk_{\mathrm{CH}}^M, T_l) = vk_{\mathrm{id_C}}$
$k \leftarrow$ *private storage*
If $T_m < T_l$: OUTPUT 0
$r'_0 := r'$; $vr := 0$; $t := \mathrm{CHF}(h_1(m||N||cnt), x)$
$r'_{cnt} := h_2(k||cnt)$
$t_{cnt} := pk_{\mathrm{CH}}^M \cdot g^{r'_{cnt}}$;
If $t_{cnt} = t$:
  $vr := 1$; $T_l := T_m$
OUTPUT vr

Figure 8: The Modified Verification Algorithms.

修改后的算法将无法提供公开可验证性，只能由一组受信任的验证者进行验证

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

LiS 应用场景

- 卫星导航系统：

  GPS和GNSS等依赖卫星的定位系统可以提供导航和时间同步功能，通过发起欺骗攻击，攻击者可以有效地迫使 GPS/GNSS 接收器得出错误的定位和导航解决方案，LiS1 可用于验证 GPS/GNSS 信号和消息，可以使用服务器辅助补货 (SAR) 刷新卫星的验证密钥

- 海事系统：

  船舶上使用的最重要的系统之一是自动识别系统 (AIS)。 它可以显示船舶的一些信息，如唯一标识、位置、航向和速度，LiS2 可以作为一种解决方案，以标准要求的快速（即 < 27 毫秒）将经过验证的 AIS 信息广播到不同的验证者，例如其他船只、灯塔或浮标

- 关键基础设施：

  可编程逻辑控制器 (PLC) 用于直接控制物理过程。为确保 PLC 正确运行并控制流程，SCADA 验证从 PLC 获得的数据的真实性至关重要，可以将 LiS1/LiS2 集成到 PLC 的固件中，并对发送到 SCADA 的每个数据进行验证。LiS1/LiS2 也可以应用于传感器，防止通过数字通道进行传感器数据注入攻击

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

性能比较和测试

- “CMAu”和“PV”分别表示连续消息认证和公共可验证性。 并且让‘SKR’表示补充签名者密钥（即是否需要补充签名密钥），‘VKR’表示补充验证者密钥
- 使用“A”、“M”、“E”、“I”、“U”和“H”分别表示加法、标量乘法、求幂、求逆、通用哈希函数和哈希函数
- ‘SEUF-(w)CMA’代表强存在性不可伪造性（弱）选择消息攻击

Table 3: Comparison

	Properties					Key Size		KGen	Computation Cost		Signature Size
	Security	SKR	VKR	CMAu	PV	Signer	Verifier		Signer	Verifier	
Schnorr [9]	SEUF-CMA	×	×	√	√	$1\|\mathbb{Z}_q\|$	$1\|\mathbb{Z}_p\|$	1E	1A+1M+1E+1H	2E+1H	$1\|\mathbb{Z}_q\|+1\|\mathbb{Z}_p\|$
Γ-1 [12]	SEUF-CMA	√	×	×	√	$2\ell \cdot \|\mathbb{Z}_p\|$	$(\ell+1) \cdot \|\mathbb{Z}_p\|$	$\ell \cdot (1E + 1M + 1H)$	1M+1A+1H	1E+1M+1H	$1\|\mathbb{Z}_q\|+1\|\mathbb{Z}_p\|$
Γ-2 [12]	SEUF-CMA	√	×	×	√	$3\ell \cdot \|\mathbb{Z}_p\|$	$(2\ell+1) \cdot \|\mathbb{Z}_p\|$	$\ell \cdot (1E + 1M + 1H)$	1M+1A+1H	1E+1M+1H	$1\|\mathbb{Z}_q\|+1\|\mathbb{Z}_p\|$
SEMECS [13]	SEUF-CMA	×	√	×	√	$\|\mathbb{Z}_q\|$	$(2\ell+1)\|\mathbb{Z}_q\|$	$\ell(4H + 1E)$	1A+1M+3H	2E+3H	$2\|\mathbb{Z}_q\|$
$\text{LiS}_1^{\text{SAR}}$	SEUF-wCMA	×	√	√	√	$5\|\mathbb{Z}_q\|$	$1.44\ell \cdot \epsilon+1\|\mathbb{Z}_p\|$	ℓ(2E+1BF+1U)+1E	3A+2M	2E+1BF	$1\|\mathbb{Z}_q\|$
$\text{LiS}_1^{\text{VSR}}$	SEUF-wCMA	×	×	√	×	$5\|\mathbb{Z}_p\|$	$1\|\mathbb{Z}_p\|+3\|\mathbb{Z}_q\|$	3E+1U	3A+2M + ℓ·U	2E+ 1H	$1\|\mathbb{Z}_q\|$
$\text{LiS}_2^{\text{SAR}}$	SEUF-CMA	×	√	√	√	$4\|\mathbb{Z}_q\|$	$1.44\ell \cdot \epsilon+1\|\mathbb{Z}_p\|$	ℓ(2E+1BF+1H)	3A+2M+2H	2E+1H+1BF	$1\|\mathbb{Z}_q\| + \ell_r$
$\text{LiS}_2^{\text{VSR}}$	SEUF-CMA	×	×	√	×	$4\|\mathbb{Z}_p\|$	$1\|\mathbb{Z}_p\|+3\|\mathbb{Z}_q\|$	3E	3A+2M+2H	2E+2H	$1\|\mathbb{Z}_q\| + \ell_r$

Schnorr : Claus-Peter Schnorr. 1989. Efficient Identification and Signatures for Smart Cards. In CRYPTO. Springer, 239–252

Γ-1 and Γ-2 : Andrew Chi-Chih Yao and Yunlei Zhao. 2013. Online/Offline Signatures for Low-Power Devices. IEEE Trans. Information Forensics and Security 8, 2 (2013), 283–294.

SEMECS : A. A. Yavuz and M. O. Ozmen. 2019. Ultra Lightweight Multiple-time Digital Signature for the Internet of Things Devices. IEEE Transactions on Services Computing (2019), 1–1. https://doi.org/10.1109/TSC.2019.2928303

用于网络物理系统中连续消息认证的轻量级签名方案

- 表 2 中报告的所有基准测试结果都是在服务器端的 Intel Core i7-4770K 和客户端的 Raspberry Pi 3 上获得的。 服务器的操作系统是Ubuntu 16.04，运行在VMWare虚拟机上，只使用了一个CPU核心

- |p| = 1024 和 |q| = 320，使用 SHA2 [22] 实现 LiS2 中的哈希函数 h。 SHA2 的实现取自高度优化的 MIRACL 库 [28]。 我们用不同的错误参数（即 10-3、10-6、10-9、10-12）对布隆过滤器进行了基准测试，最终固定参数 10−9（作为示例）以显示我们签名方案的计算成本

Table 2: The Runtime of Sign and Verify

	Sign	Verify
LiS_1^{SAR}	7.32μs	0.59 ms
LiS_1^{VSR}	7.32μs	Figure 9
LiS_2^{SAR}	11.06μs	0.59 ms
LiS_2^{VSR}	11.06μs	0.35 ms

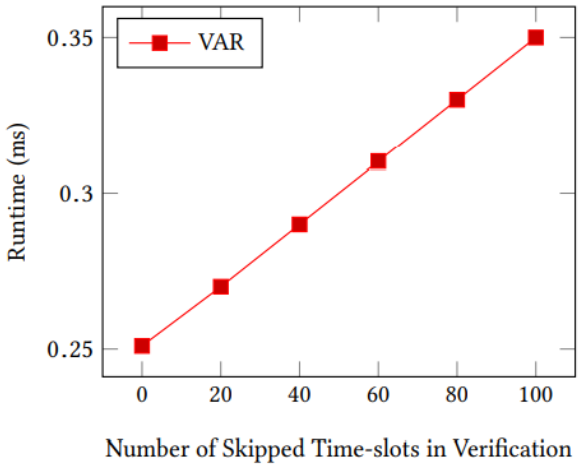


Figure 9: Runtime of LiS_1.Verify.

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.

用于网络物理系统中连续消息认证的轻量级签名方案

KGen 算法的性能如图 10 所示:

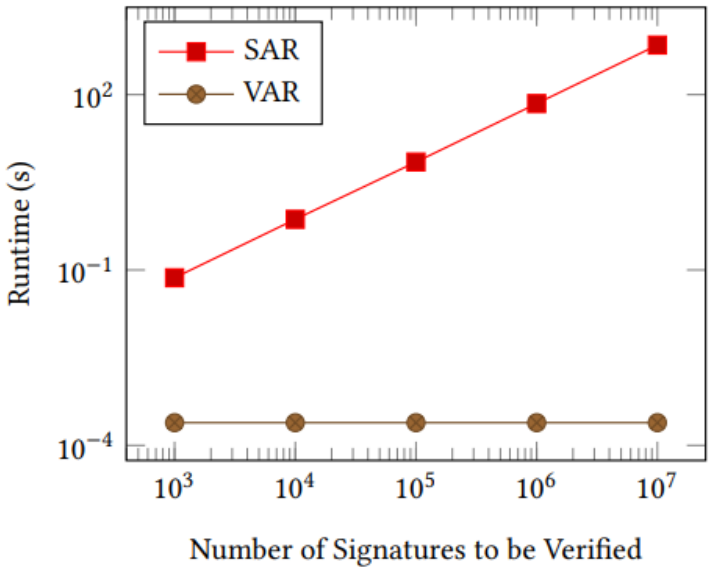


Figure 10: Runtime of KGen.

各种参数下的验证密钥的大小：

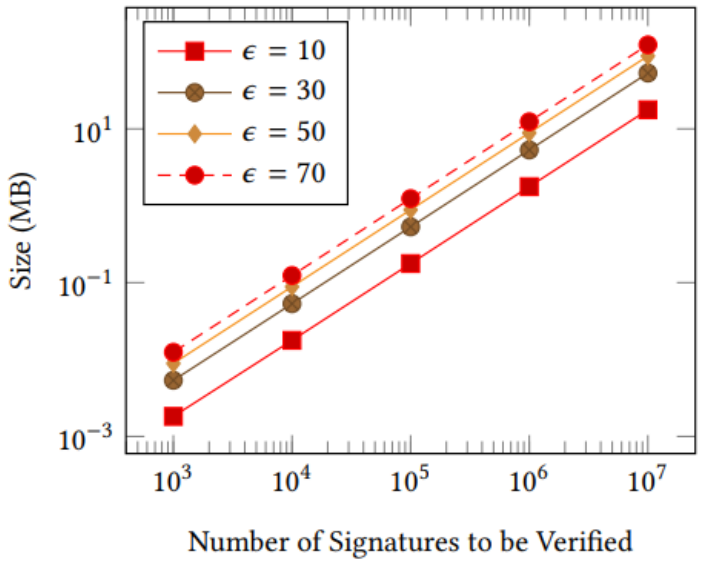


Figure 11: Size of Verification Key of LiS with SAR.

Yang Z, Jin C, Tian Y, et al. LiS: lightweight signature schemes for continuous message authentication in cyber-physical systems[C]//Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020: 719-731.