武漢大學

基于格的两方协同签名

报告人：文嘉明

学号：2022102210019

2022/11/14

抗量子数字签名

量子计算机的发展对传统的公钥密码方案构成了潜在的威胁，美国国家标准与技术研究院（NIST）在2016年启动抗量子计算密码的国际标准征集项目，经过三轮评估，于2022年7月确定了将会标准化的算法，其中最为推荐的数字签名算法是基于格困难问题设计的CRYSTALS-Dilithium.

中国密码学会也在2019年举办了全国密码算法设计竞赛，唯一获得一等奖的数字签名方案Aigis-sig也基于格困难问题，且采用了和Dilithium相似的结构.

多方协同签名

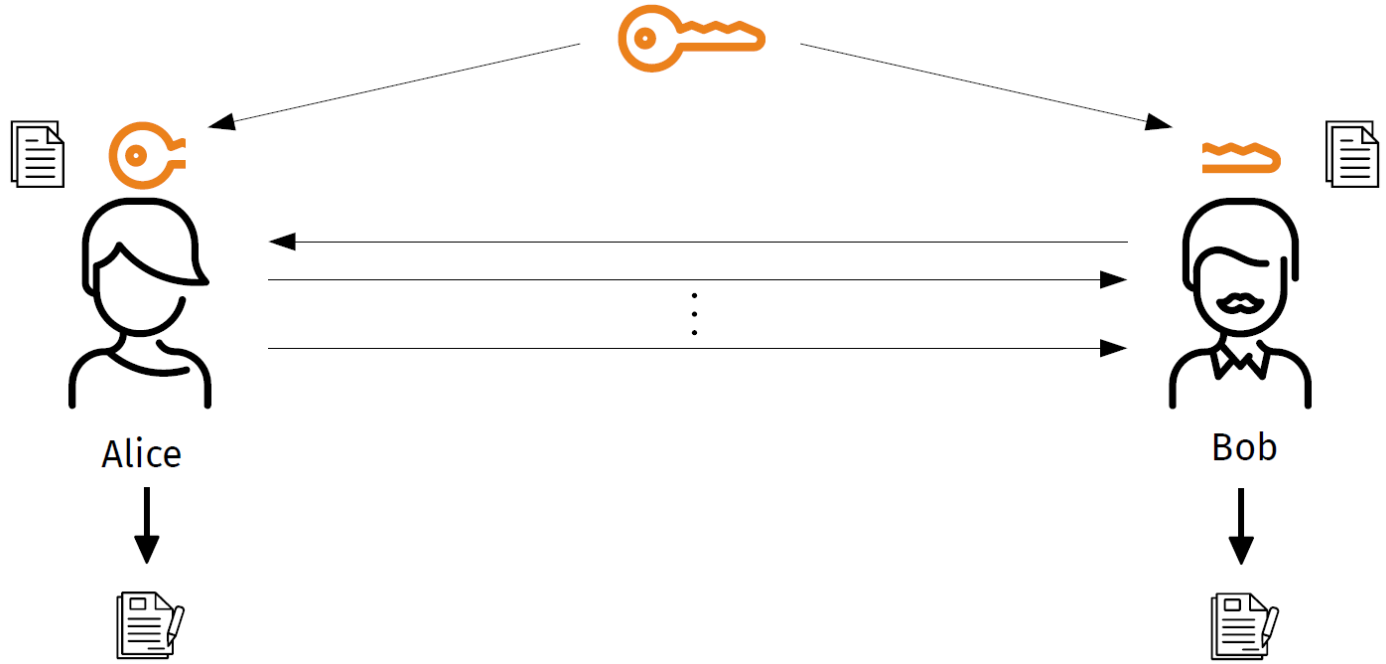
物联网和区块链等技术的兴起和发展，使得多方协同签名重新受到了关注. 多方协同签名是一种特殊的数字签名，要求多个用户各自保存私钥的一部分，在进行交互后生成公共的公钥，在对一个消息进行签名时也需要交互，以达到消息完整性和认证的目的.

优点：

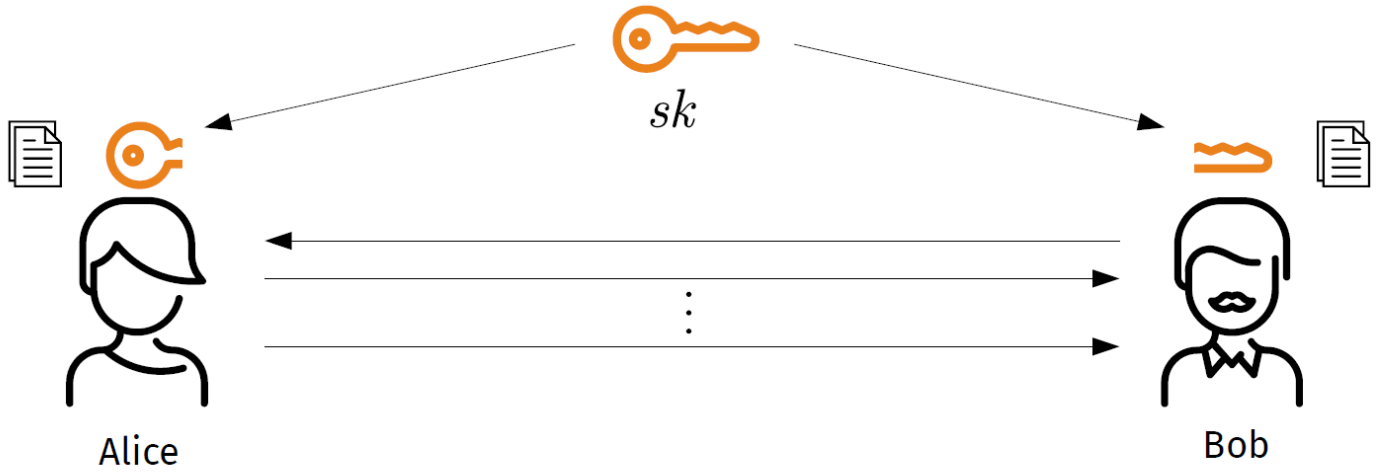
1. 相比起每个用户分别进行签名可以缩短尺寸.
2. 使用分布式的方法，任何一方都无法独自进行签名，防范因为单个用户的密钥丢失或被劫持而导致被冒充身份的隐患.

签名及验证过程(以两方为例)

签名过程



验证过程



Correctness

Verify(, , pk)=1

相关工作

经典：尽管多方协同签名协议已经被研究了很长时间，然而现有的大多数工作都集中在多方协同RSA签名、多方协同ECDSA签名、多方协同Schnorr签名以及多方协同SM2签名等，这些方案都基于整数分解或者离散对数困难假设，Shor已经证明了这些困难假设无法抵抗量子计算机的攻击.

抗量子：2019年，Cozzo和Smart对NIST征集的所有进入第二轮的签名算法转换成多方协同签名进行了评估，得出的结论是如果直接使用已有的安全多方计算通用技术，基于格困难问题的方案将需要用到线性秘密共享和混淆电路等，以及之间的互相转换，会带来较大的计算开销、需要较长的时间.

相关工作

2021年，Damgard等人利用Lyubashevsky等人提出的构造基于格的数字签名的“FSwA”结构，以及（陷门）加法同态承诺得到了两个基于格的两方协同签名，可以看作Dilithium签名方案的分布式版本，文中给出了完整的安全性证明，其安全性基于格密码中的MSIS/MLWE问题.

Vakarjuk等人也给出了一个三轮的两方协同签名方案-Dilizium，相比起上面的方案，不同之处在于使用SWIFFT同态哈希函数替换加法同态承诺方案，虽然得到了更小的密钥和签名尺寸，但是其缺点在于依赖于Rejected MLWE困难假设，是一种启发式的困难假设.

此外，这两个方案都没有考虑到Dilithium用到的签名尺寸压缩技巧，因此签名尺寸较大. 目前最新方案Dilizium 2.0在Dilizium的基础上，采用压缩技术，更接近于NIST标准化的Dilithium算法，然而该工作并没有评估实际的效率（重复次数）、密钥和签名尺寸等，也未进行参数选取和实例化.

本文贡献

1. 利用非对称模格问题，将Dilizium 2.0两方协同签名方案进行了修改和推广，能更灵活地选择参数，从而在安全性、计算效率、密钥和签名长度三方面达到更好的权衡，综合性能更优.
2. 使用“FSwA”范式构造基于格的多方签名在安全性上需要解决的一个关键问题是防止未通过拒绝抽样时的私钥信息泄露，我们使用同态承诺解决了这个问题. 对于新设计的两方协同签名方案，我们提供了完整的安全性证明，结果表明其可以有效保护各方的签名密钥，具备两方协同签名在选择消息攻击下的存在性不可伪造性.
3. 为了对比效果，给出了重复次数、密钥和签名大小的计算公式，并使用Dilithium和Aigis-sig的参数进行评估. 相比之下，本文方案的密钥以及签名尺寸优于现有的所有基于Dilithium的两方协同签名方案，例如在同等的安全强度下，签名尺寸可缩减20%以上. 据我们所知，该结果也是基于格的两方协同签名方案中最优的.

基础构造模块

承诺方案：分为承诺阶段和打开阶段，满足隐藏性(Hiding)和绑定性(Binding)两个安全属性.

承诺阶段：对于给定的消息$m \in S_m$，承诺者随机选取$r \leftarrow S_r$，并计算m的承诺值$com = Commit_{ck}(m;r)$，其中$ck \in S_{ck}$是承诺密钥，将承诺值com发送给接收者.

打开阶段：承诺者将与承诺值相关的信息发送给接收者，接收者能利用承诺值com以及相关信息计算得到一个打开值(opening)，并验证其确实是最初承诺的消息值.

隐藏性：承诺值不会泄露任何信息，即通过$com = Commit_{ck}(m;r)$无法得到关于 m 或 r 的信息.

绑定性：打开一个承诺不能得到两个不同的值，即打开$Commit_{ck}(m;r)$得到$(m';r')$，则$(m';r') \neq (m;r)$的概率是可忽略的.

基础构造模块

[ESSLL19] Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, Dongxi Liu. “Short lattice-based one-out-of-many proofs and applications to ring signatures”. ACNS 2019.

基于格的承诺方案[ESSLL19]：承诺密钥为$ck = (A_1||A_2)$，其中$A_1 = [I_k||A_1'], A_1' \leftarrow R_q^{n\times(k-n)}, A_2 \leftarrow R_q^{n\times l}$都是均匀随机选取. 在对消息$m$进行承诺时，随机选取$r \leftarrow S_r = S_\alpha^k$，得到承诺$com = A_1 \cdot r + A_2 \cdot m$. 在承诺打开阶段，用消息$m$、随机数$r$以及承诺值$com$，验证$com = A_1 \cdot r + A_2 \cdot m$ 且 $||(r,m)||_2$不超过某个阈值，若均通过则返回 1 (接受承诺)，否则返回 0 （拒绝承诺）.

高低位分解算法$Decompose_q(\theta,\lambda)$[Dilithium]：输入整数$\theta \in Z_q$和一个小的正整数$\lambda$，满足$\lambda|(q-1)$，按照下面3步操作将$\theta$分解得到$\theta = \theta_H \cdot \lambda + \theta_L$，其中$0 \le \theta_H < \frac{q-1}{\lambda}$且$||\theta_L||_\infty \le \frac{\lambda}{2}$. 并将$\theta_H = HighBits_q(\theta,\lambda)$称为$\theta$的高位比特，$\theta_L = LowBits_q(\theta,\lambda)$称为$\theta$的低位比特.

1. 将θ取$mod\ q$，落到区间$0 \le \theta < q$中，得到$\theta := \theta\ mod^+ q$.
2. 将1中的将θ取$mod\ \lambda$，落到区间$-\frac{\lambda}{2} \le \theta < \frac{\lambda}{2}$（或$-\frac{\lambda-1}{2} \le \theta < \frac{\lambda-1}{2}$）中，得到$\theta_L := \theta\ mod^\pm \lambda$.
3. 如果$\theta - \theta_L = q-1$，则令$\theta_H := 0, \theta_L := \theta_L - 1$，否则令$\theta_H := (\theta - \theta_L)/\lambda$，输出$(\theta_H, \theta_L)$.

密钥生成算法

和Dilithium算法一样，用户$P_i(i=1,2)$在选取随机的矩阵A_i时，可以直接使用256 bits的种子.

1. P_1随机选取$A_1 \leftarrow R_q^{k\times l}$并计算得到哈希值$g_1 = H_1(A_1)$，将$g_1$发送给$P_2$, P_2进行同样的操作，将$g_2 = H_1(A_2)$发送给P_1.
2. P_1收到P_2发送的g_2后，将A_1发送给P_2, P_2进行同样的操作，将A_2发送给P_1.
3. P_1收到P_2发送的A_2后，验证$H_1(A_2) = g_2$，若不成立则中止，否则计算公共矩阵$\bar{A} := [A||I_k]$，其中$A = A_1 + A_2$.

---

4. P_1随机选取$sk_{1,1} \leftarrow S_\eta^l, sk_{1,2} \leftarrow S_{\eta'}^k$，并计算得到$t_1 = \bar{A}\cdot sk_1 = A\cdot sk_{1,1} + sk_{1,2}$ 以及Hash值$g_1' = H_2(t_1)$,将g_1'发送给P_2, P_2也进行同样的操作，将$g_2' = H_2(t_2)$发送给P_1.
5. P_1收到P_2发送的g_2'后，将t_1发送给P_2, 也进行同样的操作，将t_2发送给P_1.
6. P_1收到P_2发送的t_2后，验证$H_2(t_2) = g_2'$，若不成立则中止，否则计算$t = t_1 + t_2$.

如果上述过程都没有中止，则可以得到P_i的私钥为$sk_i = (A, t_i, sk_{i,1}, sk_{i,2})$，公钥为$pk = (A, t)$.

$P_1(sk_{1,1}, sk_{1,2}; \boldsymbol{A}, \boldsymbol{t}_1, \boldsymbol{t}_2)$		$\mathrm{P}_2(sk_{2,1}, sk_{2,2}; \boldsymbol{A}, \boldsymbol{t}_1, \boldsymbol{t}_2)$
随机选择 $\boldsymbol{y}_1 \leftarrow\$ S^l_{\gamma-1}$, $\boldsymbol{r}_1 \leftarrow\$ \mathcal{S}_r = S^\kappa_\alpha$		随机选择 $\boldsymbol{y}_2 \leftarrow\$ S^l_{\gamma-1}$, $\boldsymbol{r}_2 \leftarrow\$ \mathcal{S}_r = S^\kappa_\alpha$
计算 $\boldsymbol{w}_1 := \boldsymbol{A}\boldsymbol{y}_1$		计算 $\boldsymbol{w}_2 := \boldsymbol{A}\boldsymbol{y}_2$
$\boldsymbol{w}_{1,\mathrm{H}} := HighBits_q(\boldsymbol{w}_1, 2\gamma')$		$\boldsymbol{w}_{2,\mathrm{H}} := HighBits_q(\boldsymbol{w}_2, 2\gamma')$
挑战值计算阶段:		
计算 $\boldsymbol{com}_1 \leftarrow Commit_{\boldsymbol{ck}}(\boldsymbol{w}_{1,\mathrm{H}}; \boldsymbol{r}_1)$		计算 $\boldsymbol{com}_2 \leftarrow Commit_{\boldsymbol{ck}}(\boldsymbol{w}_{2,\mathrm{H}}; \boldsymbol{r}_2)$
	$h_1 = H_3(\boldsymbol{com}_1)$ $\longrightarrow$	
	$h_2 = H_3(\boldsymbol{com}_2)$ $\longleftarrow$	
	$\boldsymbol{com}_1$ $\longrightarrow$	
	$\boldsymbol{com}_2$ $\longleftarrow$	
验证 $H_3(\boldsymbol{com}_2) = h_2$, 不成立则中止		验证 $H_3(\boldsymbol{com}_1) = h_1$, 不成立则中止
计算 $\boldsymbol{com} := \boldsymbol{com}_1 + \boldsymbol{com}_2$		
计算 $c := H_0(\mu, \boldsymbol{com}) \in \mathcal{C} := B_\tau$		

签名生成阶段：

计算　$\boldsymbol{z}_1 := \boldsymbol{y}_1 + c \cdot sk_{1,1}$

如果　$\|\boldsymbol{z}_1\|_\infty \geq \gamma - \beta$

或者　$\|LowBits_q(\boldsymbol{w}_1 - c \cdot sk_{1,2}, 2\gamma')\|_\infty \geq \gamma' - \beta'$

重新回到第一步开始计算，否则得到　$\boldsymbol{z}_1$

$\boldsymbol{z}_1, \boldsymbol{r}_1$ →

与左边类似，得到　$\boldsymbol{z}_2$

← $\boldsymbol{z}_2, \boldsymbol{r}_2$

计算　$\boldsymbol{w}_{2,\mathrm{H}} := HighBits_q(\boldsymbol{A}\boldsymbol{z}_2 - c\boldsymbol{t}_2, 2\gamma')$

如果　$Open_{ck}(\boldsymbol{com}_2; \boldsymbol{w}_{2,\mathrm{H}}; \boldsymbol{r}_2) \neq 1$，中止

计算　$\boldsymbol{z} := \boldsymbol{z}_1 + \boldsymbol{z}_2, \boldsymbol{r} := \boldsymbol{r}_1 + \boldsymbol{r}_2$

如果　$\|LowBits_q(\boldsymbol{A}\boldsymbol{z} - c\boldsymbol{t}, 4\gamma')\|_\infty \geq 2(\gamma' - \beta')$

重新回到第一步开始计算，否则计算

$\boldsymbol{w}_{\mathrm{H}} := HighBits_q(\boldsymbol{A}\boldsymbol{z} - c\boldsymbol{t}, 4\gamma')$

$\widetilde{\boldsymbol{w}}_{\mathrm{H}} := \boldsymbol{w}_{1,\mathrm{H}} + \boldsymbol{w}_{2,\mathrm{H}} \bmod \dfrac{(q-1)}{2\gamma'}$

$\boldsymbol{h} := \boldsymbol{w}_{\mathrm{H}} - \widetilde{\boldsymbol{w}}_{\mathrm{H}}$

输出签名　$\sigma := (\boldsymbol{z}, c, \boldsymbol{h})$

验证算法

收到消息 μ 对应的签名$\sigma = (z, c, h)$ 以及用来计算承诺值的r，如下验证：

1. 验证$||z||_\infty < 2\gamma - 2\beta$.
2. 用公钥$pk = (A, t)$和消息 μ 计算消息对应的承诺密钥$ck \leftarrow H_{ck}(\mu, pk)$.
3. 计算$w \coloneqq Az - ct$，从而得到高位比特$w_H \coloneqq HighBits_q(Az - ct, 4\gamma')$和$\widetilde{w_H} \coloneqq w_H - h \; mod \; \frac{q-1}{2\gamma'}$.
4. 计算$com \coloneqq Commit_{ck}(\widetilde{w_H}; r)$.
5. 验证$c = H_0(\mu, com)$，若通过则返回 1 (接受签名)，否则返回 0 （拒绝签名）.

方案的正确性

方案的正确性实际上等价于如下三个结论，我们在论文中给出了详细的推导.

1. $||z||_\infty < \gamma - \beta$.
2. $w_H := HighBits_q(Az - ct, 4\gamma')$.
3. $Open_{ck}(com, \widetilde{w_H}, r) = 1$.

方案的安全性

我们在论文中沿用Lindell给出的两方协同签名的基于Game的安全性定义[Lin17]，使用8个Hybrid Game，并结合分叉引理(Forking Lemma)，证明了本文提出的两方协同签名方案满足分布式签名的选择消息攻击下的存在性不可伪造性(DS-EU-CMA)，其安全性可以归约到非对称模格问题AMLWE/AMSIS.

[Lin17] Yehuda Lindell. “Fast secure two-party ECDSA signing”. CRYPTO 2017 & Journal of Cryptology 2021

性能分析与比较

我们使用Dilithium和Aigis的参数进行实例化，结果如下：

单位: Bytes

方案	公钥大小	每个用户的私钥大小	签名大小
Aitps-1024(本文)	2752	3328	2404
Dilizium 2.0-1024	3008	3904	3236
Dilizium-1024	3008	4032	5406
Aitps-1280(本文)	3584	4480	3140
Dilizium 2.0-1280	3744	4896	4196
Dilizium-1280	3744	5024	6750
Aitps-1536(本文)	4288	5216	3876
Dilizium 2.0-1536	4480	5536	5156
Dilizium-1536	4480	5632	8094

目前格密码方案的主要瓶颈在尺寸，而不是效率.

结论：

1. 公钥和签名尺寸更优，其中签名尺寸能缩减超过20%.
2. 期望重复次数略大，但由于Aigis在进行工程上的优化以及使用AVX2进行加速后，具体实现时甚至在一些参数集上快于Dilithium，我们合理的认为实际情况下本文方案与对比方案效率相当.

方案	期望重复次数	方案	期望重复次数
Aitps-1024(本文)	$e^{3.534} = 34.3$	Aigis-sig-1024	5.86
Dilizium 2.0-1024	$e^{3.495} = 33.0$	Dilithium-1024	5.9
Aitps-1280(本文)	$e^{4.0575} = 57.8$	Aigis-sig-1280	7.61
Dilizium 2.0-1280	$e^{3.763} = 43.1$	Dilithium-1280	6.6
Aitps-1536(本文)	$e^{3.791} = 44.3$	Aigis-sig-1536	6.67
Dilizium 2.0-1536	$e^{2.908} = 18.3$	Dilithium-1536	4.3


THANK
YOU!