基于格的密码学技术专题讲座(三)

第 6 讲　基于格的密码学在信息安全上的应用

王兆丽，杨　明，韩敬利，董　会
（解放军理工大学指挥信息系统学院，江苏 南京 210007）

摘　要：文章从数字签名、数据外包计算与安全多方计算三方面介绍了基于格的密码学在信息安全上的应用，详细阐述了将格密码应用于在不同领域的基本思想和具体实现方案。

关键词：格密码；信息安全；数字签名；外包计算；安全多方计算
中图分类号：TP309.2　　**文献标识码**：A　　**文章编号**：CN32-1289(2014)03-0100-05

Application of Lattice-based Cryptology on Information Security

WANG Zhao-li, *YANG Ming*, *HAN Jing-li*, *DONG Hui*
(College of Command Information System, PLAUST, Nanjing 210007, China)

Abstract: This article described the application of lattice-based cryptology on information security in the field of digital signature, outsourcing computation and secure multi-party computation with specifical explanation of the design strategy and the construction of the schemes.

Key words: Lattice-based cryptology; information security; digital signature; outsourcing computiong; secure multi-party computation

随着信息技术的不断发展，信息安全已成为维持社会稳定与发展的关键因素，密码技术是信息安全的理论基础和核心。随着量子计算的发展，解决密码学传统困难问题的量子算法相继出现，这给密码学带来了严峻的挑战，可抵抗量子算法攻击的后量子密码系统的研究得到广泛关注，而基于格的密码系统是抗量子攻击密码很具代表性的一种。随着近年来对基于格的密码学的研究逐渐深入，其在信息安全方面的应用正在出现。本文主要介绍其在数字签名、数据外包计算和密文检索等方面的应用。

1　格密码在数字签名中的应用

公钥密码的一个最重要应用是构造数字签名算法，随着信息技术的发展，对数字签名提出了越来越多的新需求，因而产生了一些特殊的签名方案，例如盲签名、代理签名、环签名和群签名等。这些特殊的签名方案被广泛应用于电子现金和电子投票等领域，具有很强的实用性。专题第 2 讲[1]介绍了由 Gentry 等提出的称为 GPV 的基于格的签名方案(Gentry-Peikert-Vaikuntanathan 签名方案)，本节介绍几种应用基于格的密码原语构建的特殊签名方案。

*　收稿日期：2014-05-20；修回日期：2014-06-20
作者简介：王兆丽(1980－)，女，硕士.

专题第 2 讲介绍了由 Alwen 和 Peikert 提出的基于随机格的陷门生成算法，为了引用方便，将该算法命名为 TrapSamp 算法。接下来介绍了 Gentry 等人提出的基于随机格构造的原像采样陷门函数方案 LPSF，该方案由 TrapGen(1^n)、Eval($\boldsymbol{A}$, $\boldsymbol{x}$)和 SamplePre($\boldsymbol{A}$, $\boldsymbol{T}$, s, $\boldsymbol{u}$)三个算法构成。文中还会用到文献[2]中提出的一种正交采样算法 SuperSamp(1^n, 1^m, q, $\boldsymbol{B}$)：其中 $q \geqslant 2$, $m \geqslant n+8n\log q$, $\boldsymbol{B} \in \mathbf{Z}^{n\times m}$ 算法输出 $\boldsymbol{A} \in \mathbf{Z}_q^{n\times m}$ 和 $\boldsymbol{T} \in \mathbf{Z}_q^{n\times m}$，其中 $\boldsymbol{A}\boldsymbol{B}^{\mathrm{T}} = 0 \bmod q$, $\boldsymbol{A}$ 是接近于 $\mathbf{Z}_q^{n\times m}$ 的均匀分布，$\boldsymbol{T}$ 的列向量是 $\Lambda^{\perp}(\boldsymbol{A})$的一组基，即有 $\boldsymbol{A}\boldsymbol{T} = 0 \bmod q$。

1.1　盲签名

盲签名由 Chaum 在 1982 年提出[3]，是消息拥有者在不让签名者获取他所签署消息具体内容的情况下所采取的一种特殊的数字签名技术。盲签名允许消息拥有者先将消息盲化，再让签名者对盲化的消息进行签名，最后消息拥有者对签名除去盲因子，得到签名者关于原消息的签名。关于盲签名有一个直观的比喻：先将隐蔽的文件放进信封里，当文件在一个信封中时，任何人不能读它，对文件签名就是通过在信封里放一张复写纸，签名者在信封上签名时，他的签名便透过复写纸签到文件上，而除去盲因子的过程就是打开信封的过程。通常盲签名要具有不可伪造性、不可抵赖性、盲性和不可跟踪性。盲签名可应用于电子现金和电子投票等领域，以电子现金为例说明盲签名的应用：人们在用实际钞票消费时，不会在钞票上写上自己的名字，同样道理，在使用电子现金时，当然不希望银行通过追踪自己发出的签名，来获得用户的消费情况，于是就可以采用盲签名。下面介绍一种基于格的两轮盲签名方案[4]。

设 n 为安全参数，签名者以一个 n 维格 Λ 的一组好基 $\boldsymbol{B}$ 作为签名密钥。$\boldsymbol{B}'$ 为基 $\boldsymbol{B}$ 的施密特正交基，$\boldsymbol{A} \in \mathbf{Z}_q^{n\times m}$ 是格 Λ 的校验矩阵，b 和 b' 是格 Λ 上两个离散高斯分布的参数。$h:\{0,1\}^n \to \mathbf{Z}_q^n$ 是一个安全的哈希函数。M 为要签名的消息。

方案 1　基于格的盲签名方案

第一步：消息盲化。计算 $\boldsymbol{H} = h(M)$，从以零为中心的离散正态分布随机选择向量 $\boldsymbol{c} = \{c_1, c_2, \cdots, c_m\}$，再随机选择 $t \in \mathbf{Z}$，使得 $1 < t < \|\boldsymbol{B}'\| - 1$，计算 $\boldsymbol{\mu} = (t^{-1}\boldsymbol{H} + \boldsymbol{A}\boldsymbol{c}) \bmod q$，$\boldsymbol{\mu}$ 即为盲化后的消息。

第二步：签名。签名者利用原像采样陷门函数(PSFs)生成 $\boldsymbol{\mu}$ 的签名，首先随机选择向量 $z \in \mathbf{Z}_q^m$，使得 $\boldsymbol{A}z = \boldsymbol{\mu}$，以 $-z$ 为中心，在参数 b' 下利用抽样算法抽取向量 $\boldsymbol{v}$，然后计算 $\boldsymbol{e}' = \boldsymbol{v} + z$，将 $\boldsymbol{e}'$ 作为消息 $\boldsymbol{\mu}$ 的签名。

第三步：去盲。计算 $\boldsymbol{e} = t(\boldsymbol{e}' - \boldsymbol{c})$，$\boldsymbol{e}$ 即为原始消息 M 的签名。

第四步：验证。计算 $\boldsymbol{H} = h(M)$，如果 $\boldsymbol{A}\boldsymbol{e} = \boldsymbol{H} \bmod q$，且 $\|\boldsymbol{e}\| \leqslant b\sqrt{m}$，则接受签名，否则拒绝。

可以证明，该签名方案具有盲性和 one-more 不可伪造性。

盲签名在某种程度上保护了参与者的利益，但不幸的是盲签名的匿名性可能被犯罪份子所滥用。为了阻止这种滥用，人们又引入了公平盲签名的概念。公平盲签名比盲签名增加了一个特性，即建立一个可信中心，通过可信中心的授权，签名者可追踪签名。

1.2　群签名

群签名的概念由 Chaum 和 van Heyst 在 1991 年提出。在群签名方案中，群由群管理员创建，并生成主公钥和私钥，每个授权的群成员拥有自己的由主私钥生成的私钥，他可以使用自己的私钥代表整个群体对消息进行签名，任何主公钥的拥有者能够验证签名的有效性，但不能判断具体是哪个成员实施的签名，而只有群管理员可以追踪到签名的实施者。下面介绍 Gordon 等人提出的一种基于格的群签名方案，该方案利用原像采样陷门函数 PSFs 生成追踪密钥，利用正交抽样的方法分配群成员的签名密钥。

设 n 为安全参数，$q = \mathrm{poly}(n)$，$m \geqslant 8n\log q$，$a = \omega(\sqrt{n\log q \log n})$，$\omega(\sqrt{n\log n})$ 为超级对数，即 $\omega(\sqrt{n\log n})$ 比 $\sqrt{n\log n}$ 随 n 增长更快。$h:\{0,1\}^n \to \mathbf{Z}_q^n$ 是一个安全的哈希函数，令 $\mathrm{dist}(\Lambda\boldsymbol{A}^{\mathrm{T}}, z) = \min_{s\in\mathbf{Z}_q^n} \|(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{s} - z) \bmod q\|$。

方案 2　基于格的群签名方案

第一步：密钥生成。群管理员调用 N 次 TrapSamp(1^n, 1^m, q)算法生成($\boldsymbol{B}_1$, $\boldsymbol{S}_1$), …, ($\boldsymbol{B}_N$, $\boldsymbol{S}_N$)，然后

计算 SuperSamp(1^n, 1^m, q, $\boldsymbol{B}_i$)，输出($(\boldsymbol{A}_i, \boldsymbol{B}_i)_{i=1}^N$)作为系统公钥，$(\boldsymbol{S}_i)_{i=1}^N$作为追踪密钥，$(\boldsymbol{T}_i)_{i=1}^N$作为群成员的签名密钥。

第二步：签名。假设群成员 j 要对消息 M 签名，首先随机选择 $\gamma \leftarrow \{0, 1\}^n$，令 $\overline{M}=M \| \gamma$，对每个 $1 \leqslant i \leqslant N$，计算 $\boldsymbol{H}_i=h(\overline{M} \| i)$，然后计算 $\boldsymbol{e}_i \leftarrow$SamplePre($\boldsymbol{A}_j$, $\boldsymbol{T}_j$, a, $\boldsymbol{H}_j$)；对每个 $i \neq j$，随机选择 $\boldsymbol{e}_i \in \boldsymbol{Z}_q^m$ 使得 $\boldsymbol{A}_i \boldsymbol{e}_i = \boldsymbol{H}_i \bmod q$。

对每个 i，抽样 $\boldsymbol{s}_i \leftarrow \boldsymbol{Z}_q^n$ 并且计算 $\boldsymbol{z}_i=(\boldsymbol{B}_i^{\mathrm{T}} \boldsymbol{s}_i+\boldsymbol{e}_i) \bmod q \in \boldsymbol{Z}_q^m$。最后构造一个非交互不可区分 NIWI(Non-Interactive Witness Indistinguishable)证明 χ[2]，输出签名 $\boldsymbol{\sigma}=(\gamma, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_N, \chi)$。

第三步：验证。验证者首先计算 $\overline{M}=M \| \gamma$，然后验证 χ，如果 χ 是正确的，并且对每个 $1 \leqslant i \leqslant N$，都有 $\boldsymbol{A}_i \boldsymbol{z}_i=h(\overline{M} \| i) \bmod q$，则输出 1；否则输出 0。

第四步：打开。使用追踪密钥$\{\boldsymbol{S}_i\}$，输出使得 $\mathrm{dist}(\boldsymbol{\Lambda}(\boldsymbol{B}_i^{\mathrm{T}}), \boldsymbol{z}_i)$最小且 $\mathrm{dist}(\boldsymbol{\Lambda}(\boldsymbol{B}_i^{\mathrm{T}}), \boldsymbol{z}_i) \leqslant a \sqrt{m}$的 i。

吴雍东[5]对上述方案的安全性和实用性进行了分析，指出该方案存在两点不足：首先它不能防止陷害攻击，即群管理员可以假冒任意群成员生成合法签名；其次不能灵活地增删群成员，不适合动态群。文献[5]利用统计零知识证明方法对原方案进行了改进，解决了上述问题。

1.3 代理签名

代理签名是一种特殊的数字签名，在电子商务和安全协议中有着广泛的应用。代理签名的概念由 Mambo 等在 1996 年提出[6]，其主要思想是原始签名人将他的签名权委托给代理签名者，代理签名者代表原始签名人生成数字签名，任何持有原始签名人公钥的人都可以对代理签名进行验证。一个代理签名方案通常由四个算法构成：密钥生成、代理私钥生成、代理签名和签名验证。夏峰[7]等基于格中平均情况下小整数解问题 SIS(Small Integer Solution)和非均匀小整数解问题 ISIS(Inhomogeneous Small Integer Solution)的困难性假设，构造了一种基于格的高效代理签名方案。方案利用原像采样陷门函数方案 LPSF 中的 TrapGen 算法生成原始签名者和代理签名者的公私钥对，利用盆景树下格的扩展及格基的随机化方法生成代理私钥，利用 LPSF 中的 SamplePre 算法生成代理签名。与传统的基于数论的代理签名方案相比，该方案的密钥空间较大，但运算简单(只需线性运算)，且能抵抗量子攻击。

2 格密码在数据外包计算中的应用

目前，云计算以低成本、高利用率、灵活性及良好的扩展性等诸多优点迅速被学术界和 IT 界所认同。云计算的实质是数据存储和计算外包，数据存放、处理由第三方负责，除了对数据控制的担忧之外，人们还对云计算可能导致的隐私问题忧心忡忡。用户不愿意将数据以不加密的形式存放在云服务器中，特别是一些隐私数据，但是将数据加密又丧失了云计算的优势，除非云计算能够在不知道密钥、不解密的情况下来处理数据。全同态加密技术为这一难题提供了良好的解决方案，它将隐私保护和计算便利有效的结合在一起，云端服务器不用解密就可以处理敏感数据[8]。

2.1 代理计算

当前，拥有较弱计算资源的用户将计算委托给计算能力强大的云来处理的趋势越来越明显。同时，智能手机、笔记本等移动设备的普及，这些设备也需要外包计算。同时，委托给云的计算如此重要以至于必须排除在计算当中出现偶然误差的可能。但是他们必须面对这样一个现实：很多种恶意攻击可对网络中的设备以及其上的数据造成伤害，提供云计算服务的服务商会有很大的诱因返回错误的答案。在这些情况下，用户需要能够利用较少的计算资源、有效的验证服务器返回结果的正确性。下面介绍一种基于全同态加密技术的可验证的代理计算方案[9]。

方案 3 基于全同态加密技术的可验证的代理计算方案

离线阶段：D（代理者）和 W（服务器）均可收到计算函数 F。

D 生成全同态加密的密钥对 $(p, s)\leftarrow\mathrm{keyGen}(1^k)$（$k$ 为安全参数）。计算 t 个独立的密文 $\hat{r}_i=\mathrm{Enc}_p(\overline{0})$，然后进行同态密文计算 $\hat{w}_i=\hat{F}(\hat{r}_i)=\mathrm{Eval}_p(\hat{r}_i, F)$，将 (p, s)，$(\hat{r}_i, \hat{w}_i)$ 保存为秘密（其中 $i\in[t]$）。

在线阶段：D 和 W 均可收到计算输入 $x\in\{0, 1\}^n$。

第一步：D 计算 t 个独立密文 $\hat{r}_{i+t}=\mathrm{Enc}_p(x)$，其中 $i\in[t]$，取样一个随机的排列 $\pi\in_{\mathrm{R}}\mathbf{S}_{2t}$。令 q 表示 $(p, \hat{z}_{\pi(1)}, \cdots, \hat{z}_{\pi(2t)})=(p, \hat{r}_1, \cdots, \hat{r}_{2t})$。然后 D 生成一组新的密钥对 $(p', s')\leftarrow\mathrm{keyGen}(1^k)$，将 p' 和 $\hat{q}=\mathrm{Enc}_{p'}(q)=\mathrm{Enc}_{p'}(p, \hat{r}_1, \cdots, \hat{r}_{2t})$ 发送给 W。

第二步：W 利用全同态加密方案的 Evaluate 算法同态计算元组 $\mathrm{Enc}_{p'}(\hat{y}_1, \cdots, \hat{y}_{2t})=\mathrm{Enc}_{p'}(\hat{F}(\hat{r}_1), \cdots, \hat{F}(\hat{r}_{2t}))$ 并发送给 D。

第三步：D 解密 W 的信息得到 $(\hat{y}_1, \cdots, \hat{y}_{2t})$，然后验证两个部分：首先，验证 $\hat{w}_i=\hat{y}_{\pi(i)}$，其中 $i\in[t]$。然后解密 $\hat{y}_{\pi(i+t)}$，其中 $i\in[t]$，看解密结果是否一致。如果两步验证均通过则接受和输出解密的结果。

该方案在即使服务器知道计算输入、计算函数的情况下，仍能进行正确的验证。

2.2　加密信息检索

当用户为了保护隐私将数据加密后存放在云服务器中时，如何对加密后的数据进行检索就成了我们要面临的新问题。加密信息检索可以看做是代理计算的一个特例。使用全同态加密技术处理加密信息检索具有先天的优势，因为全同态加密方案可以对加密数据进行任意函数的计算，所以理论上可以处理任意检索请求，包括模糊匹配、包含任意逻辑连接词的关键词检索等。

全同态加密可支持即使不知道私钥的用户也可以对文件进行检索，但是返回的检索结果无法解密，没有实际意义。下面介绍的加密检索方案[10]仍然是拥有私钥的共享用户对数据进行检索，在检索过程中，检索要求和关键字都是加密状态，信息不会泄漏。

方案 4　基于全同态加密技术加密信息检索方案

第一步：数据拥有者 Sender 从密钥中心获取全同态加密方案的密钥对 (p, s)，同时数据的共享者（数据拥有者授权的用户）也获得私钥。

第二步：利用全同态加密方案 E，数据拥有者使用公钥对文件和文件的关键词进行加密 $\psi_i=\mathrm{Encrypt}_E(p, w_i)(i=1, \cdots, t)$，然后上传到云服务器。

第三步：共享用户 Reciver 有检索要求，首先从密钥中心获取自己的密钥对 (p', s')。

第四步：用户 Reciver 将检索请求和陷门上传到云服务器。检索请求：检索线路 C（包含检索词密文 $\psi_i'=\mathrm{Encrypt}_E(p', w_i')(i=1, \cdots, t)$ 和逻辑联接词信息），陷门：$\bar{s}=\mathrm{Encrypt}_E(p', s)$。

第五步：服务器首先利用公钥 p' 将文件的关键词进行二次加密 $\overline{\psi}_i=\mathrm{Encrypt}_E(p, \psi_i)(i=1, \cdots, t)$，然后利用全同态加密方案中的 Evaluate 算法进行检索操作 $y=\mathrm{Evaluate}_E(p', C, \overline{\psi}_i, \bar{s})$。如果 y 值为 1，则文件满足检索要求，否则不满足。

3　安全多方计算

安全多方计算 SMC(Secure Multi-party Computation) 主要研究网络环境下多个互不信任参与方的协作计算问题，使拥有私有数据的多个参与者能够合作利用这些私有数据进行计算，同时又不泄露各自私有数据的机密。安全多方计算是信息安全近年来的研究热点之一，其在电子商务、军事等领域都有广泛的应用。

同态加密技术是安全多方计算的核心技术之一。在基于格的密码方案中，NTRU 因其具有加法同态和混合乘法同态的特性，被用于设计安全多方计算协议[11]。而基于理想格的全同态加密方案因其完美的全同态特性，已成为解决安全多方计算的强有力技术手段。使用这一技术，每个参与者可以先使用全同态方案加

密输入数据，然后使用同态属性计算想要的函数值，最后分别解密密文以得到最终的结果。

在设计安全多方计算协议的过程中，参与者的行为会决定协议的设计难度，下面以半诚实参与者为例进行说明。半诚实参与者会严格执行协议过程，不与其他参与方合谋，但可能会保留所有中间结果，并试图从这些结果中推导出协议之外的信息。假设有两个半诚实的参与者 Alice 和 Bob，基本的协议[12]过程如下。

协议 1 基于半诚实模型的安全两方计算协议

Alice 的输入：$a\in\{0,1\}^n$，Alice 发送。

Bob 的输入：电路 $C:\{0,1\}^n\rightarrow\{0,1\}$。

第一步：Alice 生成全同态加密的私钥和公钥对(e,d)，然后将 e 发给 Bob。

第二步：Alice 发送 $\boldsymbol{x}=(\mathrm{Encrypt}_e(x_1),\mathrm{Encrypt}_e(x_2),\cdots,\mathrm{Encrypt}_e(x_n))$给 Bob。

第三步：Bob 计算 $\boldsymbol{c}=\mathrm{Evaluate}_e(C,\boldsymbol{x})$，然后将其发给 Alice。

第四步：Alice 使用私钥 d 解密 $\boldsymbol{c}$，得到她的输出。

下面讨论如何将协议 1 从两方协议扩展成 3 方协议。首先将 Alice 拆分成 Alice1 和 Alice2，他们的输入分别为 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$，Alice1 的目标得到 $C(\boldsymbol{x}_1,\boldsymbol{x}_2)$，这里的 C 是 Bob 的输入电路。Alice1 和 Alice2 运行一个两方安全协议共同生成一个全同态加密的私钥和公钥对(e,d)，随机生成 d_1 和 d_2，使得 $d_1\oplus d_2=d$。Alice1 得到 d_1，Alice 得到 d_2，然后她们分别加密各自的输入并将结果发给 Bob，Bob 分别计算 Evaluate 并将结果返回给她们。Alice1 和 Alice2 再运行一个安全两方协议解密 Bob 返回给他们的数据，最后把结果告知 Alice1。如果要将协议扩展成 n 方，可重复刚才的整个过程。

4 结束语

在过去的十几年中，基于格的密码学因其基于最坏情况难解问题的安全性、抗量子攻击和高效等优点得到了广泛关注，正是由于格密码所具有的这些优点，促使学者们在进行格密码基础理论研究的同时，不断尝试将其应用于信息安全领域的实践中。文章从数字签名、数据外包计算和安全多方计算三方面介绍了基于格的密码学在信息安全上的应用，详细阐述了将格密码应用于在不同领域的基本思想和具体实现方案。

参考文献：

[1] 韩敬利，杨 明，王兆丽. 基于格的密码函数构造及其应用[J]. 军事通信技术，2014，35(1)：75-80.

[2] Dov G S，Katz J，Vaikuntanathan V. A group signature scheme from lattice assumptions[C]// Asiacrypt 2010. Singapore，Berlin：Springer-Verlag，2010：395-412.

[3] Chaum D. Blind signatures for untraceable payments[C]// Crypto'82. Santa Barbara：Springer，1982：199-203.

[4] 王凤和，胡予濮，王春晓. 基于格的盲签名方案[J]. 武汉大学学报：信息科学版，2010，35(5)：550-553.

[5] 吴雍东. 基于格的群签名方案[J]. 小型微型计算机系统，2011，32(11)：2243-2247.

[6] Mambo M，Usuda K，Okamoto E. Proxy signatures for delegating signing operation[C]// ACM Conference on Computer and Communications Security. New York：ACM Press，1996：48-57.

[7] 夏 峰，杨 波，马 莎，等. 基于格的代理签名方案[J]. 湖南大学学报：自然科学版，2011，38(6)：84-88.

[8] Lauter K，Naehrig M，Vaikuntanathan V. Can homomorphic encryption be practical?[EB/OL]. (2011-05-10)[2014-04-23]. http：// www. codeproject. com/News/15443/Can-Homomorphic-Encryption-be-Practical. aspx.

[9] Gennaro R，Gentry C，Parno B. Non-interactive verifiable computing：outsourcing computation to untrusted workers[C]// Crypto'10，Santa Barbara：Springer，2010：547-565.

[10] Boneh D，Gentry C，Halevi S，et al. Private database queries using somewhat homomorphic encryption[C]// ACNS'10. Banff，Cannda：Springer-Verlag，2013：102-118.

[11] Sheikha R，Mishra D K，Kumar B. Secure multiparty computation：from millionaires problem to anonymizer[J]. A Global Perspective，2011，20(1)：25-33.

[12] Du W L，Atallah M J. Secure multiparty computation problems and their applications：a review and open problems[C]// ACM Symposium on New Security Paradigms Workshop. New York：ACM Press，2001：11-20.