关于数据库与隐私同态

Ronald L. Rivest

Len Adleman

Michael L. Dertouzos

麻省理工学院

马萨诸塞州剑桥市

一、引言

加密是一种众所周知的保护敏感信息隐私的技术。这项技术一个基本且看似固有的限制是，处理加密数据的信息系统最多只能为用户存储或检索数据；任何更复杂的操作似乎都需要先解密数据才能进行。然而，这种限制源于所使用的加密函数的选择，尽管在可实现的目标上存在一些真正固有的限制，但我们将看到，对于许多有趣的操作集合，似乎很可能存在一些加密函数，允许在未预先解密操作数的情况下对加密数据进行操作。我们将这些特殊的加密函数称为“隐私同态”；它们构成了任意加密方案（称为“隐私变换”）的一个有趣子集。

作为一个示例应用，考虑一家使用商用分时服务存储记录的小型贷款公司。该贷款公司的“数据库”显然包含应保密的敏感信息。另一方面，假设分时服务采用的信息保护技术不被贷款公司认为足够充分。特别是，系统程序员可能会访问到敏感信息。因此，贷款公司决定对其数据库中保存的所有数据进行加密，并坚持仅在总部解密的政策——数据绝不会在分时计算机上解密。这种情况如图1所示，其中波浪线环绕的区域代表贷款公司的物理安全场所。

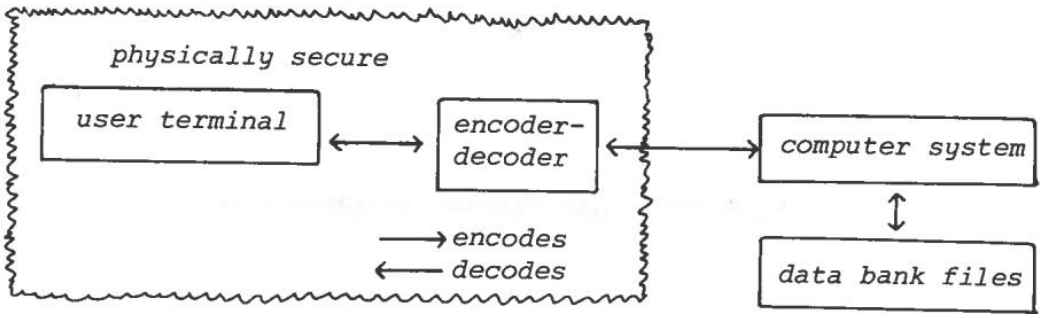

图1

这种组织方式允许贷款公司利用分时服务的存储设施，但通常使得在不损害存储数据隐私的情况下利用其计算设施变得困难。然而，贷款公司希望能够回答诸如以下问题：

- 未偿还贷款的平均规模是多少？
- 下个月预计有多少贷款收入？
- 已发放了多少笔超过5000美元的贷款？

这些问题的回答需要进行计算。

贷款公司可能采取四种方案：

（1）放弃使用分时服务的想法，购买内部计算机系统。
（2）仅使用分时服务的存储设施来存储加密数据，并在贷款公司办公室使用一个“智能终端”进行必要的解密和计算。
（3）说服分时公司对其计算机进行硬件修改，使得数据能够在其CPU内部以解密形式短暂存在，但解密后的数据无法从外部访问。
（4）使用一种特殊的隐私同态对其数据进行加密，使得分时计算机能够操作数据而无需先解密。

方案（1）可能非常昂贵，且不一定能解决问题——可能需要某种形式的加密来保护存储的信息免受内部系统程序员的盗窃或恶意篡改。方案（2）可行，但通常会产生相当大的通信成本。方案（3）也可行，但需要分时公司的合作。在第2节中，我们将简要讨论此方案。方案（4）只需要存在合适的隐私同态，并且贷款公司获得实现该同态的加密/解密设备即可。在第3至5节中，我们将分别探讨这种解决方案的数学要求、其适用性的一些限制以及一些可能有用的隐私同态。

二、通过硬件修改的解决方案

在图2中，我们展示了计算机系统如何修改以安全地执行加密数据操作的示意图。除了标准的寄存器组和算术逻辑单元（A, B）外，增加了一组物理安全的寄存器组和算术逻辑单元（C, D）。主存储器与物理安全寄存器组之间的所有数据通信都经过一个配备用户密钥的编码器-解码器（E），因此未加密的数据只能存在于物理安全的寄存器组内。主存储器、数据库文件、普通寄存器组以及通信信道上的所有敏感数据都将被加密。在操作期间，主存储器与安全寄存器组之间的加载/存储指令将自动执行相应的解密/加密操作。

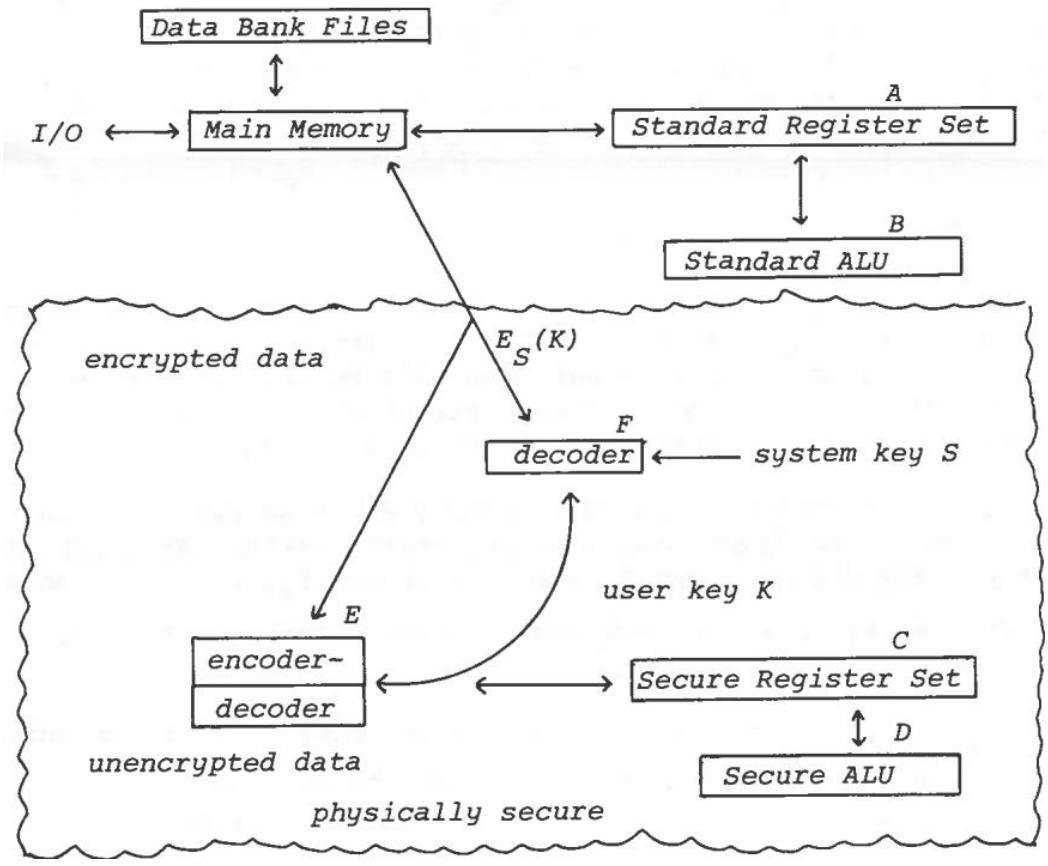

图2

一个明显的问题是如何在不泄露用户密钥K安全性的情况下，将编码器/解码器（E）加载用户的密钥K。一种可能的方法是，将用户的密钥K在系统密钥S的控制下保持加密状态。K的加密形式 $E_{S}(K)$ 可以通过不安全信道传输到系统，由物理安全的解码器（F）解密，并加载到编码器-解码器（E）中。用户知道K和 $E_{S}(K)$；后者是在访问分时服务经理时获得的，经理是唯一知道系统密钥S的人。

除了密钥管理问题外，还存在每次加载或存储时调用加密/解密导致速度下降的问题。然而，看起来适当安全的加密（例如DES）可以在许多机器指令执行的时间尺度（例如10微秒）内完成。

然而，此解决方案最严重的限制，结果证明也是任何解决问题方案（甚至是隐私同态）的一个限制：不可能同时保持安全性并赋予系统执行与已知常量进行比较操作的能力。也就是说，我们不能赋予计算机系统执行足够强大操作的能力，使得仅知道 $\mathbf{E}_{\mathrm{S}}(\mathbf{K})$ 的人能够解密数据。执行与常量比较的能力将允许某人执行简单的二分搜索程序来确定任何数据的解码值。我们将在第4节更详细地探讨此限制。

三、隐私同态

人们可能更倾向于一种不需要解密用户数据（当然在用户终端除外）的解决方案。也就是说，硬件配置将如图1所示，但使用的加密函数将允许计算机系统在不解密数据的情况下对其进行操作。

我们假设未编码的数据以及要在其上执行的操作来自某个代数系统。一个代数系统由一个集合S、一些运算 $f_1, f_2, \ldots$ 、一些谓词 $p_1, p_2, \ldots$ 和一些特定常量 $s_1, s_2, \ldots$ 组成。我们将此系统表示为 $<S; f_1, f_2, \ldots; p_1, p_2, \ldots; s_1, s_2, \ldots>$ 。例如，由整数及其常规运算集组成的系统可以表示为 $<Z; +, -, x, \div; \leq; 0, 1>$ ；其中Z是整数集。

除了用户的代数系统（我们称之为U）外，我们还需要另一个代数系统C供计算机系统使用。编码和解码则意味着将元素从U映射到C或反之。更正式地，如果

那么

并且我们必须有一个解码函数 $\phi: S' \to S$ 及其逆函数，即编码函数 $\phi^{-1}: S \to S'$。

在实际操作中，用户向计算机系统提供代数系统C的描述；实际上，这意味着系统有一个子程序来计算每个运算 f_i' 和谓词 p_i'，以及特定常量 s_i' 的表示。用户的实际数据库我们表示为序列 $d_1, d_2, \ldots$，每个 d_1 是S的一个元素。然而，用户在将每个数据项交给系统之前会对其进行编码；编码后的数据库是 $\phi^{-1}(d_1), \phi^{-1}(d_2), \ldots$。

为了使系统能够在（编码后的）数据库上进行操作而无需解密，解码函数 ϕ 必须是从C到U的同态映射。形式上，这意味着

并且

ϕ 将C中的每个运算映射到U中对应的运算。假设现在用户想知道 $f_{1}(d_{1}, d_{2})$ 的值。他要求系统计算 $f_{1}'(\phi^{-1}(d_{1}), \phi^{-1}(d_{2}))$ 。由于 ϕ 是同态，

因此，系统得到了答案的加密形式，而无需解密中间结果。一般而言，任何使用U的运算来计算用户数据库某个函数的计算机程序，都可以通过将所有 f_{i} 替换为 f_{i}'、所有 p_{i} 替换为 p_{i}'、所有 s_i 替换为 s_i' 的方式，转换为适合在编码数据上操作的另一计算机程序。

对代数系统 C 和函数 ϕ, ϕ^{-1} 的选择要求如下：

（1）解码和编码函数 ϕ 和 ϕ^{-1} 应易于计算。
（2）C中的运算 $f_{i}^{\prime}$ 和谓词 $p_{i}^{\prime}$ 应可高效计算。
（3）数据 d_{i} 的编码版本 $\phi^{-1}(d_{i})$ 所需的表示空间不应比 d_{i} 的表示空间大太多。
（4）知道许多数据 d_i 对应的 $\phi^{-1}(d_i)$ 不应足以揭示 ϕ。（唯密文攻击）
（5）知道多个 d_i 及其对应的 $\phi^{-1}(d_i)$ 不应揭示 ϕ。（选择明文攻击）
（6）C中的运算和谓词应不足以高效计算 ϕ。（这主要与比较运算的使用有关）。

四、一些简单的观察

一些固有的限制限制了我们所描述的隐私同态的效用。最严重的可能是以下这一点。

事实：如果C中可用的运算允许计算机系统确定任意常量的编码版本，并且存在用于全序的谓词“ $\underline{\underline{\mathbf{\Pi}}}$ ”，则不存在从C到U的安全隐私同态。

这源于一种简单的“二分搜索”策略。例如，对于自然数系统

和

对于某个集合 W，计算机系统上的恶意系统程序员可以通过计算 $\phi^{-1}(1) = 1'$、$\phi^{-1}(2) = 1' + 1'$、$\phi^{-1}(4) = \phi^{-1}(2) + \phi^{-1}(2)$ 等等，直到找到满足 $\phi^{-1}(2^k) \geq \phi^{-1}(d_i)$ 的k。继续下去，类似的策略使他能够精确计算出 d_i。

关于一个系统模拟另一个系统的能力的其他事实不那么容易看出，但可以被发现。例如，我们有如下事实。

事实：如果 C 基于自然数，并且具有加法、乘法运算，一个二元相等谓词和一个一元谓词“等于”零，那么它就具有测试是否等于任意常量的能力。

Lynch [1] 对能够模拟 $+$ 的一个代数系统与另一个代数系统之间的关系进行了出色的研究。

五、一些示例隐私同态

我们在此给出四个示例隐私同态。这些主要是作为例子，用以支持在许多应用中可能存在有用隐私同态的假设。其中一些在密码学上相当弱；“选择明文攻击”可能破解它们。我们仍然列出它们，以说明可能存在的隐私同态类型。

和减法运算，其中p是素数。我们可以选择 $C = \langle Z_n; x_n, \div_n \rangle$，即模n的整数，其中 $n = p \cdot q$，是p和一个大素数q的乘积。设g是模p的一个生成元。那么我们选择

并且解码函数是基于 g 的“模(p)对数”逆函数。根据指数定律，ϕ 是一个同态。如果 n 难以分解（p 和 q 都很大）并且素数 p 满足可以高效计算模 p 的对数（参见[2]），那么计算机系统可以同时拥有 g 和 n，而不用担心危及数据安全。

示例 2：假设 $\mathrm{U} = <\mathrm{Z}_{\mathrm{p}}; \mathrm{x}_{\mathrm{p}}; \equiv_{\mathrm{p}} >$，模 $\mathfrak{p}$ 的整数，带有乘法和相等性测试。再次令 $n = p \cdot q$，其中 q 是一个大素数，并假设 n 难以分解，我们可以取

因为 $(\mathbf{x}^{\mathrm{e}})(\mathbf{y}^{\mathrm{e}}) = (\mathbf{xy})^{\mathrm{e}}$，所以这是一个同态。实际上，这正是 Rivest、Shamir 和 Adleman 在他们实现公钥密码系统的方法[3]中使用的编码函数。该系统的安全性应该非常好，即使计算机系统同时拥有 e 和 n。

示例 3：$U = \langle Z_n; +_n, -_n, x_n \rangle$，其中 n 再次是两个大素数 p 和 q 的乘积，使得 n 难以分解。我们选择用一对数字来表示 Z_n 中的每个元素：

计算机系统通过按分量执行操作（模 n）来形成两个编码的和、差或积。在不知道 p 和 q 的情况下，系统无法解码任何数字。由于给定数字可能有多种编码，因此无法进行相等性测试。

示例 4：假设 $U = \langle Z; +, -, x \rangle$，即整数在通常的加法、减法和乘法运算下的系统。用户选择一个整数 n 并以基数-n 表示法表示其所有数据。计算机系统可以在不知道 n 的情况下（因此也不知道未加密的数据）对这些值进行操作，允许单个坐标位置超过 n。例如，如果 $n = 17$，我们有

同样，由于一个给定数字可能有多种表示形式，无法进行相等性测试。计算机系统也可以通过仅使用该常数在个位位置来找到任意给定常量的编码。

通过组合上述两种系统，可以使用“分数”来实现有理数。该系统虽然具有许多其他良好特性，但并不能真正抵御“选择明文攻击”。

示例 5：假设我们再次有 $U = \langle Z; +, -, x \rangle$，即整数在加法、减法和乘法运算下的系统。选择 k 使得任何计算中使用的中间结果都小于 2^k，并设 $a_0, a_1, \ldots, a_{k-1}$ 是 k 个随机选择的整数。整数 x 的编码（其中 x 以二进制表示 $x = x_{k-1} \ldots x_1x_0$，每个 x_i 是 0 或 1）是 k 元组 $(f_{x}(a_0), f_{x}(a_1), \ldots, f_{x}(a_{k-1}))$，其中

编码后的表示可以按分量进行操作。解码意味着对给定值进行多项式插值，然后在点 $Z = 2$ 处评估该多项式。这种隐私同态的空间效率不高。该系统的安全性，即使对抗选择明文攻击，似乎涉及求解 a_{i} 的高阶非线性方程，但可能存在密码分析的捷径。

六、结论

隐私同态为确保必须进行操作的数据的隐私提供了一种新颖的方法。它们的适用性本质上是有限的，因为比较运算通常不能包含在要使用的操作集中。此外，是否可能拥有一个具有大量操作集且高度安全的隐私同态仍有待观察。本文给出的结果为对寻找有用的隐私同态提供了一些乐观的基础；这里给出的示例虽不非常实用，但具有启发性。悬而未决的问题是：

- 这种方法是否具有足够的实用性，使其在实践中值得采用？
- 对于哪些代数系统 U 存在有用的隐私同态？

参考文献

[1] Lynch, N. and E. Blum, "Efficient Reducibility Between Programming Systems", Proc. 9th Annual ACM Symposium on Theory of Computing, (Boulder, May 1977), pp. 228-238.

[2] Pohlig, S. and M. Hellman, "An Improved Algorithm for Computing Logarithms Over GF(p) and Its Cryptographic Significance" (to appear IEEE Trans. Info. Theory).

[3] Rivest, R., A. Shamir, and L. Adleman, "A Method For Obtaining Digital Signatures and Public-Key Crypto Systems", Massachusetts Institute of Technology, Laboratory for Computer Science, Technical Memo, TM-82, April 1977. (to appear in CACM).

讨论

Rabin：我想提一个关于安全性的额外考虑。这里一个最吸引人的提议实际上是进行模算术。当 n 是 p 和 q 的乘积，并且你进行分量方式的模算术时，你不是分别对模 p 和模 q 进行，而是在模 n 算术中进行。这看起来相当不错，因为我们不知道如何分解数字。然而，破解任何这些系统的一种可能性是，对手拥有特殊知识。有时对手控制着部分输入数据。他是一家银行中的储户，而银行正在操作他的银行账户。因此，他实际上知道正在被编码的 A、B、C 等值。现在，必须考虑到通过查看编码，可能能够在这种情况下找到 n 的因子分解。

Rivest：那么很可能有人能够破解它。

Rabin：是的，在评估其安全性时，必须考虑到通过向其提供已知信息并跟踪其在加密版本中的过程，或者向其提供加密信息并跟踪其过程来挑战系统的类似考虑。

Gaines：也许我误解了，但我认为 p 和 q 将在系统外选择。为每个个体单独选择。这样就没有人有机会做你所说的事情。

Rabin：我可以补充一点吗？如果你提议为每个客户使用不同的 p 和 q，这相当困难且不切实际，有时一个不无辜的旁观者知道你存了多少钱。那么另一个问题仍然存在。你必须至少假设对将要保护的数据有零星的部分了解。

Rivest：我认为我提出的所有系统都容易受到此类攻击的变体影响。正是由于这些原因，我认为它们都不令人非常满意。

专业术语中英文对照表

英文术语	中文翻译
Data Bank	数据库
Privacy Homomorphism	隐私同态

英文术语	中文翻译
Privacy Transformation	隐私变换
Encryption/Decryption	加密/解密
Time-sharing Service	分时服务
Intelligent Terminal	智能终端
Hardware Modification	硬件修改
Encoder/Decoder	编码器/解码器
Physically Secure Register Set	物理安全寄存器组
Arithmetic Logic Unit (ALU)	算术逻辑单元 (ALU)
Key Management	密钥管理
System Key	系统密钥
Algebraic System	代数系统
Set	集合
Operation	运算
Predicate	谓词
Distinguished Constants	特定常量
Homomorphism	同态
Encoding/Decoding Function	编码/解码函数
Ciphertext Only Attack	唯密文攻击
Chosen Plaintext Attack	选择明文攻击
Total Order	全序
Binary Search	二分搜索
Modulo Arithmetic	模算术
Generator (modulo p)	生成元 (模p)
Modulo Logarithm	模对数
Factorization	因子分解/因式分解
Public-Key Cryptosystem	公钥密码系统
Componentwise	按分量/分量方式
Radix Notation	基数表示法
Polynomial Interpolation	多项式插值
Binary Notation	二进制表示法