加密数据库技术：前沿与展望

任 奎[1] 王 聪[2]
[1]浙江大学
[2]香港城市大学

关键词：加密数据库 可搜索加密 可信硬件

加密数据库体系及核心需求

数据是驱动数字经济发展最核心动力，此观念已然成为行业共识。以数据为基础的大数据、云计算、物联网、区块链、人工智能等数据科学在智慧城市升级、国家重大基建产业发展等方面发挥着重要的作用。考虑到数据作为核心生产要素的重要地位，保护数据的安全和隐私不容忽视。它关乎国家安全，尤其数据科学与工业生产的相互融合使得数据安全的影响蔓延到军事、金融、医疗、教育等各个领域。在此背景下，学界和工业界都已经开始大力推动大数据安全战略布局，各国政府也都相继出台各项法律法规以规范保障数据的安全使用和生产，如我国的《网络安全法》、《密码法》等。

然而，近年来频频暴发的数据泄露事件表明，数据安全保护的现状仍不容乐观。据 IBM 安全机构发布的消息称，仅在过去的一年里，由于数据泄露事件造成的平均经济损失高达 386 万美元[1]。为了最大程度防止数据隐私泄露，保障数据在整个生命周期内（包括：存储、传输以及处理 / 运行）的安全性成为迫切需要解决的问题。对于数据存储和传输过程中的安全保护，已经有了诸多受到业界认可的国内外安全标准和算法，比如 AES、国密及 TLS 等。这些技术的使用可以极大降低数据在静态存储及流转时的风险。

但是对于数据运行时的保护，却仍存在很大的局限性。随着数据成为不可或缺的核心生产要素，其在生产使用过程中的相关安全隐患也逐渐显露出来。具体来说，在诸多应用场景中，程序运行时内存中的数据仍然是以明文方式存在的，给了来自内部 / 外部攻击者可乘之机。为了保护运行时的数据安全，就要求做到数据“可用”但“不可见”。这不仅有助于抵御来自上述内部 / 外部的攻击者，达到数据的纵深安防保护，也能够尽最大可能地保留数据作为生产要素的原生价值[2]。

目前，数据全生命周期的安全保护（尤其是运行时安全）是当今数据安全行业内（学界和工业界）公认的热点问题。随着应用场景对数据安全要求的不断提高，数据运行时安全的技术方向和发展趋势日新月异，涌现出了像同态加密、安全多方计算、可搜索加密、可信硬件等优质的安全防护技术。这些热点技术侧重点各异，正在学界和工业界的一同推进下飞速发展。不仅有来自谷歌、微软等开源库的分享，也有国内阿里牵头，针对安全多方计算国际标准的推动。这些努力极大推进了数据安全建设的步伐，维护了数据安全生产的稳定性。

本文拟从推动数据运行时安全的战略需求出发，聚焦加密数据库这一前沿领域，深入介绍相关可行

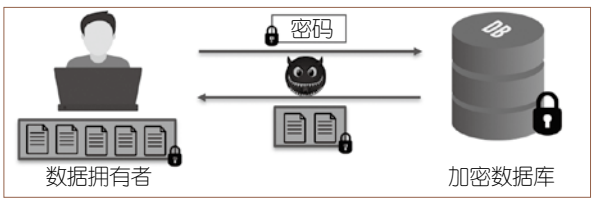

图1 基于关键词的可搜索加密示例

性技术路线的发展现状，包括可搜索加密技术以及近年来趋于成熟的可信硬件技术等。与此同时，我们也将针对相关技术路线所面临的不足和挑战，展望潜在研究方向，并探讨关于加密数据库建设的一些未来愿景。

可搜索加密的诞生与发展

如图1所示，可搜索加密主要解决的是加密数据库的安全检索问题。其主要过程如下：（1）数据拥有者在本地加密数据并上传到加密数据库服务器。（2）数据库服务器在不可见“查询请求明文”和“数据内容明文”的情况下，仍然能够进行“加密”查询操作，并将匹配的结果（密文）集返回给用户。

实现这类可搜索加密的密码学工具种类繁多，大致包括：（1）属性保护加密算法（Property-Preserving Encryption，PPE），包括保序加密（Order-Preserving Encryption，OPE；Order-Revealing Encryption，ORE）、确定性加密（Deterministic Encryption，DTE）等；（2）同态加密（Fully / Somewhat Homomorphic Encryption，HE）；（3）不经意随机访问机制（ORAM）；（4）功能加密（Functional Encryption）；（5）对称可搜索加密（Searchable Symmetric Encryption，SSE）。在上述各种安全设计中，PPE在功能和效率上有突出的表现，却泄露了更多信息；ORAM、HE能提供较强的安全性保证，但相应的性能方面仍然有待加强。相比之下，SSE能平衡效率和安全性。

基于对称密钥的可搜索加密

对称可搜索加密（SSE）[3]的设计思想来源于在明文上基于索引的搜索机制。给定明文数据集，数据拥有者首先构建基于明文关键词/文档的逆序搜索索引（inverted index），该索引记录与关键词相匹配的文件集合。例如：$K_1 \rightarrow \{F_1, F_4, F_2\}$记录了包含关键词$K_1$的文档集合$\{F_1, F_4, F_2\}$。然后，利用单向函数以及对称密钥工具，将该搜索索引加密，生成一个可以搜索的加密索引结构，同时也对数据集进行加密。加密索引的内容并不涉及数据本身的明文信息，仅描述了关键词与文件集的映射关系。注意到该加密索引数据结构在搜索查询之前是加密状态，仅当接收到合法授权的加密查询请求时，才能打开对应项目进行查看。在后续的查询过程中，对于任意一个搜索请求，可以通过一个安全的单向函数（如基于密钥的伪随机函数），将明文查询请求（如K_1）转变成密文请求token(K_1)。对应的加密索引项目token(K_1)被打开后，与该token对应的文件集标识1、4、2被暴露出来，这样服务器可以将加密文件F_1、F_4、F_2返回。在整个查询过程中，查询请求明文和数据内容对服务器是不可见的。

在过去的近二十年里，学界和业界的专家学者们为可搜索加密在功能、效率和安全性方面的完善付出了很多的努力，极大地推动了可搜索加密的发展进程[4-7]。针对一些前沿难点都有了相应的代表性技术方案予以支撑，诸如：（1）支持增、删、改的动态可搜索加密方案；（2）支持范围查询、k-近邻查询、多关键词查询、布尔查询、排序查询、近似查询等丰富查询功能的加密方案；（3）大规模加密数据的搜索部署，数据I/O对性能的影响等。

可搜索加密安全定义下的泄露函数及相关攻击

上述示例中，针对整个查询过程，服务器虽然看不见明文数据，但仍然可以观察到一些信息，包括：（1）搜索模式，它记录着重复的搜索查询；（2）访问模式，满足搜索查询的（加密）文件集，即查询结果。直观上，这些信息并不泄露加密的数据内容，它们可以看作是为了让服务器更快地达到搜索查询目的，而不得不牺牲掉的一些“辅助”信息。在密码学中，这些信息的具体定义源于相关的泄露函数（leakage functions）。安全的密码学方案要求除泄露函数定义的信息以外，不泄露有关数据集的额外信息，即泄露是被允许且可控的。

基于可控泄露的安全性模型自提出以来便被运用在各种可搜索加密方案中。然而，这些泄露真的可控吗？挖掘被允许的信息泄露（如搜索结果集/

访问模式等）是否可能暴露相关明文数据特征？是否会对数据本身的机密性有影响呢？ 2012 年，Islam 等人的工作首次表明了在掌握一定先验知识的情况下，敌手可以通过 SSE 中的可允许泄露的访问模式分析出相应的查询内容，开启了泄露滥用攻击（Leakage-Abuse Attack，LAA）的先河 [8]。此后，越来越多的方案被指出存在类似安全性缺陷。

1. 访问模式泄露攻击。CCS'15（ACM 计算机与通信安全大会）提出的基于查询结果长度的攻击，针对 CCS'06 提出的允许访问模式泄露的 SSE 方案的安全性，给出了第一个较为系统性的研究 [9]。

如图 2 所示，明文索引里关键词“chair”“food”“score”所对应的结果集的大小分别为 4、4、3。此时，若敌手观察到一个加密查询 token 其返回结果的数量是 3（唯一长度），则敌手可以直接推断出该 token 所对应的查询内容为“score”。同时，在已知该 token 查询结果的基础上，若敌手发现另一个加密查询返回的结果中有两个文件与“score”的查询重合，即共现文档数（co-occurrence count）为 2，则可以断定该查询的内容为“chair”。上述攻击统称为计数攻击（count attack），即利用可搜索加密访问模式中提取的搜索结果长度和多个搜索结果间共现模式的

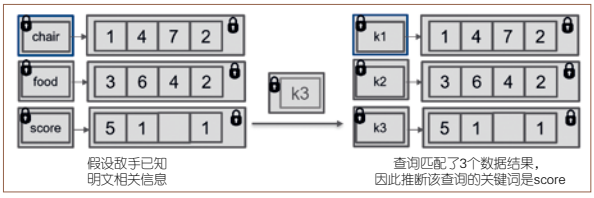

图2　基于查询结果长度的攻击示例

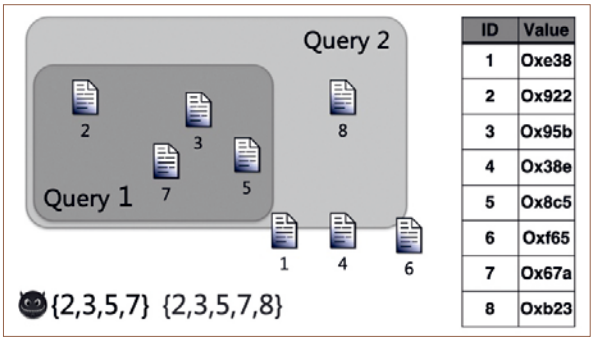

图3　针对范围查询的攻击示例

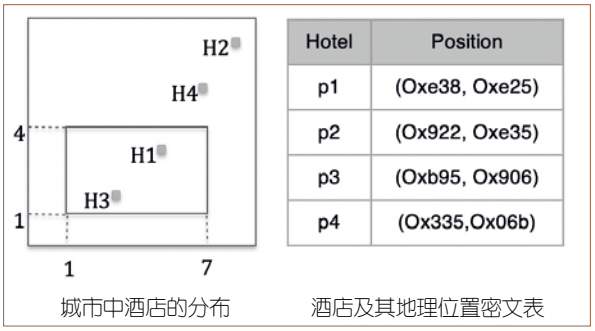

图4　针对2D范围查询的攻击示例

唯一性，作为判断依据来推断出相应的查询内容。

2. 范围查询泄露攻击。除了访问模式，可搜索加密同时还存在其他的泄露函数，它们同样可以被用来恢复查询相关的密文数据内容。比如支持范围查询（range search）的可搜索加密（例如 ORE、OPE），它们同时还泄露查询结果集合之间的顺序关系。下面介绍关于范围查询的泄露攻击示例 [10]。如图 3 所示，给定某加密表，ID 列指 row ID，Value 列指加密后的数据值，拟通过查询语句中的 where 条件范围查询来筛选相关数据。这些值加密后，都是不泄露大小关系的。搜索结果本身，比如查询请求 1（Query 1）返回的结果 {2,3,5,7} 也不泄露结果集内数据值之间的大小关系。但是查询语句中的范围查询条件“<”是有可能被允许观察到的。根据这一特性，敌手可以通过有限次的范围查询，完全或部分重构加密数据集的大小顺序。例如，敌手观测到查询请求 2（Query 2）的返回结果 {2, 3, 5, 7, 8} 包含查询请求 1 的结果集，则可以推测出文件 8 所对应的值最大。此时，假设敌手还掌握其他背景信息（比如已知部分明文数据库信息作为参考），可以进一步推测出加密数据的具体值。

3. 2D 范围查询攻击。在前期范围查询的基础上，Falzon 等人提出了针对 2D 范围内 k 邻近查询的攻击方案 [11]。顾名思义，2D 查询返回指定区域内的所有点。如图 4 左图所示，在明文情况下拟查询某城市区域 [(1,7),(1,4)] 内的酒店信息，服务器应返回酒店 H1、H3；相应的，若给定酒店位置信息的加密表（如图 4 右图所示），针对同样的范围查询，在密文情况下，假设服务器返回该查询结果

为 p2、p3。假设敌手观察到的所有结果中，返回长度为 2 的查询结果分别为 {p1, p2}、{p2, p3}、{p3, p4}。注意到，p1、p3 必然分布在 p2 的两端，否则应该存在结果 {p1, p3}，从而可以确定存在一条直线路径 p1 → p2 → p3。同理有 p2 → p3 → p4，从而推断 p1 → p2 → p3 → p4 为完整的直线路径。若背景信息表明该路径上的点对应于一条街道上的酒店，假设从远到近分别为：H3、H1、H4、H2，则敌手可以通过该顺序路径恢复加密表中数据集与明文所对应酒店间的关系。如 p1、p2、p3、p4 分别为 H3、H1、H4、H2 或者 H2、H4、H1、H3。该方法还可以拓展成支持多条路径的推断攻击。

此外，还有针对查询分布的不同假设，以及利用搜索模式来辅助攻击的方法，这里不再详述。

上述案例表明，现有的可搜索加密方案普遍存在泄露滥用攻击的安全隐患。这些隐患不仅仅影响数据的安全性，如果不给予足够的重视与防范，一旦被敌手恶意地放大利用，更可能演变成实际部署中的重大数据泄露事故。鉴于此，针对可搜索加密安全性强化的研究，逐渐成为研究者近年来关注的焦点。

可搜索加密的安全性强化与挑战

对于目前已知的攻击，学界已开始研究如何消除由这些允许的信息泄露所带来的安全性隐患。一个比较经典的设计就是使用填充（padding）来隐藏可搜索加密方案的查询结果集的数量信息（volume）[12]。如图 5 所示，中间图表示数据库中各关键词（横坐标）所对应的 volume 大小（纵坐标）。不难发现，图中大部分关键词都具有唯一的 volume 大小。而 volume 的信息在数据加密前后是不变的，因此，敌手可以通过观察各查询返回结果的 volume 大小来推断出查询的内容 [9]。

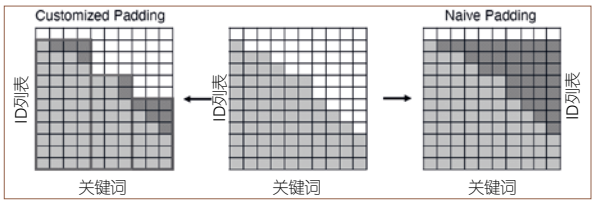

图5　针对Volume信息的填充防御示例

为了隐藏该信息，最直观的方法就是通过将虚假文件填充到原数据库中，使得所有的关键词都有相同的 volume 大小，如图 5 右图所示。这样，从 volume 的大小方面来看，敌手就无法区分不同的关键词，但是这种做法需要引入大量的虚假填充文件，会造成较大的存储和通信开销。为了解决这个问题，也有学者提出局部的填充方案，即先将关键词分组（比如每组 C 个关键词）再填充，使每一组的 C 个关键词都具有相同的 volume 大小。这样，每一组内的关键词从 volume 的角度是无法区分的。用户可以根据自己的需求选取每一组关键词的数量，以权衡相应的安全强度和存储开销（C 越大，填充越多，存储开销大，安全性越高）。

上述例子通过 volume 填充，在一定程度上缓解了针对 volume 泄露所带来的压力。但是目前加密数据库的泄露函数形式纷繁复杂，滥用攻击方式层出不穷。仅仅针对查询结果集的数量信息做防护是远远不够的，敌手仍有可能从其他角度进行攻击，比如不同关键词在同一文件内同时出现的关系。为了全面防范这类攻击，我们需要对可搜索加密的安全泄露问题有更深刻的了解。如何更全面地度量可控信息泄露的现实含义并提出相关攻防设计，将成为可搜索加密技术应用在未来实际部署过程中必须要解决的重难点问题之一。

构建加密数据库系统的探索

作为保障数据生产资料的安全基石，构建一个安全可靠的加密数据库系统有着十分重要的战略意义。可搜索加密技术作为最相关的一类密码学原语，是该安全系统研发道路上的一个重要组成部分，为数据安防保护奠定了重要的理论基础。我们在不断追求其在查询方面安全性强化的同时，也要考虑到来自系统和应用 / 业务逻辑支撑等各方面的挑战，在“丰富的查询功能”以及“高效的查询性能”方面，不断探索新的可行性技术路线。

近年来，随着可信硬件技术的发展，一些用于实现可信执行环境（Trusted Execution Environment，

TEE）的技术也趋于成熟，并逐渐进入学界和工业界的视野，例如 ARM TrustZone 和 Intel SGX。研究人员在探索如何打造一个全功能的加密数据库的技术手段上，也从早期的纯密码学方案逐渐过渡到与可信硬件相结合的方向上，拟借助可信硬件技术来构造功能和性能完备的潜在解决方案[13~17]。

基于可信硬件的加密数据库系统

Intel 推出的基于硬件的 TEE 技术称为 SGX，它允许开发者对程序进行划分，将需要保护的部分运行在一个称为 Enclave 的可信执行区域内，Enclave 外部被划分为不可信区域。硬件会保护可信执行区域内部不受其他来自不可信区域的非法访问。这样的设计让 SGX 的可信计算基（Trusted Computing Base，TCB）非常小，仅需包含硬件和运行在 Enclave 内的代码。但这导致程序执行的过程中需要在可信部分和不可信部分之间来回切换，即频繁地进出 Enclave。SGX 在物理内存中划分了一段空间用于存储 Enclave 的代码和数据。目前，该物理内存空间的上限为 256MB，SGX 使用了页交换的方式来突破物理内存大小的限制。

相比于前述基于纯密码学工具的设计思路，基于可信硬件的加密数据库拟具备如下优点，主要包括：（1）TEE 内部数据天然地具有私密性和完整性的保护；（2）TEE 能提供更丰富的功能和更好的性能，避免了复杂且功能受限的密码学方案设计。例如使用 OPE/ORE 加密方案时，仅能进行范围查询，并不能支持数据库查询中的所有运算符。而基于可信硬件的方案不需要构建复杂的密码学方案，在可信硬件内部可以直接进行安全可信的明文处理。相比之下，可信硬件能支持更复杂的加密数据查询与计算，并且性能也更优。

可信硬件给加密数据库带来灵活、高效计算性能的同时也面临着诸多问题。首先，TEE 本身存在安全性隐患，一方面是因为硬件上可能存在漏洞，另一方面是可信硬件在设计时没有考虑侧信道攻击，比如 Intel SGX 明确表明不防御侧信道攻击。虽然这些攻击需要的条件较为苛刻，但也一定程度地影响了 TEE 的安全性。其次，如果运行在 TEE 内的代码本身存在漏洞，则仍可以被攻击者利用，破坏数据库的安全性。运行在 TEE 内的代码越多，其存在漏洞的可能性越大。所以，我们希望能尽可能少地把系统模块放到 TEE 中，保证一个较小的 TCB。最后，可信硬件的使用不可避免地会引入额外开销，例如程序进出 Enclave 以及数据的页交换等。

如果希望将 TEE 应用在加密数据库上，需要克服上述缺陷，尤其是安全性和性能的问题。这与如何切分程序，即把哪部分代码和数据放到 Enclave 内紧密相关。通常，数据库由很多模块组成，包括请求的解析器、计划生成器、计划执行器、日志模块等。这些模块还可以再进一步细分，除了代码模块，还有表格、索引模块等。要把哪些模块放入 Enclave 内是利用 Enclave 实现加密数据库的一大难点问题。例如，当把计划执行器的代码、表格和索引数据放入 Enclave 时，整个查询执行的过程以及用户表中的数据都能得到保护。但是用户的查询语句以及表结构的隐私就无法顾及。把更多的代码模块和数据放入 Enclave 中可以保护更多的内容，但相应的 TCB 也将变大，同时还可能会带来性能问题。除了安全性和性能，使用 Enclave 还需要有工程难度的考量。这是由于 Enclave 对内部程序存在一定的限制，放到 Enclave 里的代码往往需要一定的修改才能够正常使用。基于这些思考，我们介绍两种代表性的设计思路示例，并分析其优缺点和可行性。

基于安全数据库管理系统的设计

基于 Enclave 实现安全加密数据库最直接的思路就是利用 Enclave 保护数据库管理系统，如 EnclaveDB[13]。此类方案一般假设 Enclave 支持的物理内

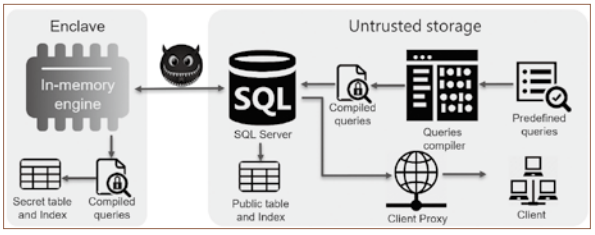

图6　基于Enclave实现加密数据库管理系统的示例

存足够大。如图6所示，基于这个假设，EnclaveDB将SQL server的内存数据库管理引擎连同所有的敏感数据，包括表和表上的索引，都保存在Enclave内存中。在每个加密表上支持的各种操作都预先由用户本地编译优化，生成编译后的查询（compiled queries），在数据库初始化阶段一同加载到Enclave中。用户的查询请求经过代理模块时，代理模块会对查询中的敏感数据加密，发送到SQL server上。如果查询是关于敏感数据的，就会通过调用保存在Enclave内的compiled queries执行查询，否则就像普通数据库一样由Enclave外部的SQL server处理。

从安全的角度来看，将完整的内存数据库引擎以及数据都放在Enclave，既保护了用户数据的私密性和完整性，又保护了数据库引擎自身的各种元数据、中间数据等信息，具有较高的安全性。从性能角度来看，数据放在Enclave内的方案，合理利用了Enclave自身的硬件加密和校验机制；同时基于Enclave内存足够大的假设，整个数据处理逻辑都在Enclave内，Enclave进出的开销与待处理的数据大小无关。但是以目前硬件的发展速度来看，要在实际应用中满足Enclave内存足够大的假设（例如支持动辄上百GB的用户数据表），仍然面临诸多技术挑战，需要学界和业界的共同努力，以及可信硬件提供商的大力投入与研发支持。

基于Enclave安全运算的设计

第二种基于Enclave的加密数据库设计，是仅利用Enclave进行安全数据运算处理[15]，与上一种设计思路相反，它旨在把尽量少的模块放入Enclave内。如图7，这类方案大多基于数据库插件扩展的

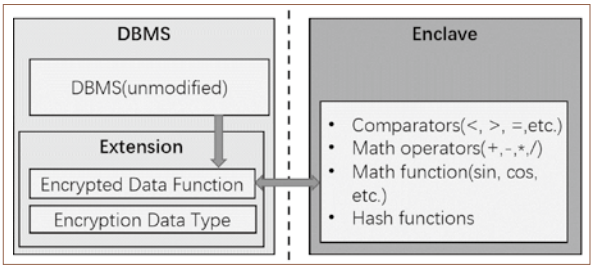

图7 基于Enclave安全运算的加密数据库示例

方式，通过扩展定义各种加密数据类型，并针对性地定制对应的表达式操作函数，例如大小比较、运算、数学函数、哈希（hash）计算等，从而实现基于Enclave的加密数据库的查询与计算。在Enclave内部的安全运算，保证了相关加密数据在运行时的安全性。这种方式可以方便地集成在现有的各种数据库上，无须对原来的数据库系统做较大的改动。

下面我们通过示例来说明基于Enclave安全运算的加密数据库处理查询请求的流程。考虑如下查询语句：`SELECT SUM(c_balance) FROM customer WHERE c_city = '0xaef5306c'`，即需要查询隶属某市的(`c_city='0xaef5306c'`，城市名是密文）所有消费者(`customer`）的账户余额(`c_balance`，余额是密文）总和(`SUM(c_balance)`)。系统首先会遍历所有的消费者，然后通过Enclave中的比较函数判断某消费者是否属于某市，获得比较结果True或者False，返回给数据库。在获得满足城市条件的数据记录之后，数据库再不断调用Enclave中的加法函数，遍历这些消费者的余额进行累加，最后计算出加密的余额总和并返回给数据库。注意到，整个查询任务执行过程中，需要进出Enclave的次数和待处理数据量有关，因此在数据量较大时，Enclave进出开销较高。

在上述过程中，由于Enclave内的数据被保护，数据库本身并不知道查询的某市到底对应哪个城市。但是因为每次Enclave的返回结果（该条目的城市是否匹配`c_city = '0xaef5306c'`）是外部可见的，这些结果泄露了加密数据之间的关系。这就意味着这种设计面临着前述可搜索加密方案中普遍存在的泄露滥用攻击（LAA）的安全隐患。若敌手有先验知识，是可能通过观察查询，推断出某些密文数据及其相关明文数据间的关联性或其他特定信息的。

从应用角度来看，上述设计方案实现简单，对诸如PostgreSQL支持插件扩展的数据库来说，只需要以插件形式扩展即可，不需要改动代码。移植到其他的数据库，也只需要非常少的修改。从性能角度看，由于Enclave内部只是加密数据表达式的计算，基本不存在内存交换，因此页交换开销非常低。但

是数据需要加解密的次数以及进出 Enclave 的次数均与待处理的数据量关联，在数据量较大时加解密开销和 Enclave 进出开销较高。在安全层面，此类设计无法像基于 Enclave 的安全数据库管理系统那样保证数据完整性。更重要的是，在查询的过程中泄露了很多加密数据关系信息。因此，其安防力度比一般使用 Enclave 的应用场景要弱，不能普适性地满足数据保护，尤其是高敏感数据的安全需求。

近期，我们课题组也对基于可信硬件的加密数据库系统进行了前沿探索。聚焦在范围查询，我们提出了一种混合索引结构的设计——HybrIDX[18]，可实现加密的范围查询，并抵御相关的各类泄露滥用攻击的安全隐患。其核心思想是依靠可信硬件，将加密比较操作移至可信赖的 Enclave 安全区，以协助安全范围查询处理，并同时降低甚至混淆不必要的信息泄露，在安全性和效率方面都有显著提升。

结语

随着数据科学的迅猛发展，保护核心生产要素的安全需求给加密数据库的构建带来了前所未有的挑战。虽然近年来可搜索加密理论飞速发展，可信硬件技术慢慢崛起，让我们看到了加密数据库领域逐步迈向成熟商业化的可能性，但将加密数据库真正落地并得到广泛应用，仍需要学界和业界的共同努力与长期探索。围绕核心安全技术攻关、系统架构标准制定以及数据库系统安全测评等方面，加密数据库的发展仍面临许多关键挑战：

1. 尽管 Enclave 在一定程度上为数据的使用提供了安全环境，但是 Enclave 和外部交互仍然会给加密数据库带来安全隐患，甚至造成潜在的数据信息泄露威胁[19]。此外，如何划分数据库模块，更好地权衡安全性、性能和工程难度，也是打造基于可信硬件加密数据库系统的一个重要难点。面对这些制约安全和性能的瓶颈，怎样进行针对性技术攻关、优化升级和改造，是扫清加密数据库推广道路上障碍的关键。

2. 已有数据库系统种类繁多，如何推进加密数据库技术的广泛部署，实现相关产品架构及其生态应用的安全升级，需要各界企事业单位以及科研院校的协同参与，开展相关标准的制定工作。不论是云数据库服务还是本地部署，都需要建立一整套成熟的架构及部署标准，从而更广泛地推进加密数据库技术落地，促进相关产业生态的发展。这也是目前需要关注与思考的重要课题。

3. 为了提升数据安全监管能力，保证数据产业的健康发展，我们还应该意识到确立一套完备的数据库安全评估体系的重要性。该评估体系应符合我国相关法律对密码产品定级以及自主可控的要求，能够有助于我们定性、定量地评估各个系统模块及相关状态下数据库的安全等级（例如查询量、泄露函数、可信硬件安全保护力度等），起到安全评估乃至实时防护警示的作用。

这些问题是推动加密数据库发展的挑战，同时也是未来研究的契机。展望未来，我们坚信经过领域同行的共同努力，加密数据库必定能够有所突破并应用到生产生活的方方面面，助力实现稳定、高效的网络空间安全以及良好生态的愿景。 ■

任　奎

CCF 高级会员。浙江大学求是讲席教授，网络空间安全研究中心主任。 主要研究方向为物联网安全、数据安全和隐私保护、人工智能安全。

kuiren@zju.edu.cn

王　聪

CCF 专业会员。香港城市大学教授。主要研究方向为数据安全、隐私保护、机密计算及其安全应用。

congwang@cityu.edu.hk

（本文责任编委：沈　超）

参考文献

[1] IBM Security.2020年度泄露报告 [R]. IBM Corporation, https://www.ibm.com/downloads/cas/BK0BB0V1, 2020.

[2] Fuller B, Varia M, Yerukhimovich A, et al. Sok:

Cryptographically protected database search[C]// *2017 IEEE Symposium on Security and Privacy* (*SP*). 2017.

[3] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: improved definitions and efficient constructions[C]// *Proc. of ACM CCS*, 2006.

[4] Wang C, Cao N, Li J, et al. Secure ranked keyword search over encrypted cloud data [C]// *Proc. of ICDCS*, 2010.

[5] Wang C, Ren K, Yu S, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data [C]// *Proc. of IEEE INFOCOM*, IEEE, 2012.

[6] Cash D, Jarecki S, Jutla C S, et al. Highly-scalable searchable symmetric encryption with support for boolean queries [C]// *Proc. of CRYPTO*, 2013.

[7]Cao, N., Wang, C., Li, M., Ren, K., and Lou, W., "Privacy-preserving multi-keyword ranked search over encrypted cloud data[J]," IEEE Transactions on Parallel and Distributed Systems, 25 (1), 222-233, 2014.

[8]Islam, M.S., Kuzu, M., and Kantarcioglu, M., "Access pattern disclosure on searchable encryption: ramification, attack and mitigation [C]," in Proc. of NDSS, 2012.

[9]Cash, D., Grubbs, P., Perry, J., and Ristenpart, T., "Leakage-abuse attacks against searchable encryption [C]," in Proc. of ACM CCS, 2015.

[10]Lacharité, M.S., Minaud, B., and Paterson, K. G., "Improved reconstruction attacks on encrypted data using range query leakage [C]," in Proc. of IEEE S&P, 2018.

[11]Falzon, F., Markatou E.A., Akshima, Cash, D., Rivkin, A., Stern, J., and Tamassia, R., "Full Database Reconstruction in Two Dimensions [C]," in Proc. of ACM CCS, 2020.

[12]Xu, L., Yuan, X., Wang, C., Wang, Q., and Xu, C., "Hardening database padding for searchable encryption [C]," in Proc. of IEEE INFOCOM, 2019.

[13]Priebe, C., Vaswani, K., and Costa, M., "EnclaveDB: A secure database using SGX [C]", in Proc. of IEEE S&P, 2018.

[14]Eskandarian, S. and Zaharia, M., "ObliDB: Oblivious query processing for secure databases[C]," in Proc. of VLDB, 2019.

[15]Vinayagamurthy, D., Gribov, A., and Gorbunov, S., "StealthDB: a scalable encrypted database with full SQL query support[C]," in Proc. of PETS, 2019.

[16]Antonopoulos, P., Arasu, A., Singh, K.D., et al., "Azure SQL database always encrypted[C]," in Proc. of ACM SIGMOD, 2020.

[17]Sun, Y., Wang, S., Li, H., and Li, F., "Building Enclave-Native Storage Engines for Practical Encrypted Databases[C]," in Proc. of VLDB, 2021.

[18]Ren, K., Guo, Y., Li, J., Jia, X., Wang, C., Zhou, Y., Wang, S., Cao, N., and Li, F., "Hybridx: New hybrid index for volume-hiding range queries in data outsourcing services [C]," In Proc. of ICDCS, 2020.

[19]Chen, Y, Li J., Xu, G., Zhou, Y., Wang, Z., Wang, C., and Ren, K., "Towards Efficiently Establishing Mutual Distrust Between Host Application and Enclave for SGX[C]," arXiv:2010.12400，2020.