(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 117874789 A
(43)申请公布日 2024.04.12

(21)申请号 202410015848.5

(22)申请日 2024.01.05

(71)申请人 北京嘉华铭品牌策划有限公司
地址 102488 北京市房山区拱辰街道月华大街1号八层815

(72)发明人 张渴欣

(74)专利代理机构 北京天奇智新知识产权代理有限公司 11340
专利代理师 龙涛

(51)Int.Cl.
G06F 21/60(2013.01)
G06F 21/62(2013.01)

权利要求书2页 说明书13页 附图1页

(54)发明名称

一种动态的隐私数据加密方法和系统

(57)摘要

本发明公开了一种动态的隐私数据加密方法和系统,所述方法包括:步骤S1:对敏感数据进行自动识别与分区;步骤S2:对数据块更新,并将数据块与MPC任务关联;步骤S3:实行动态增量加密机制;步骤S4:存储节点将增量加密后的数据应用到相关数据块中,同时保持未变更数据的加密状态不变。本发明提出的动态隐私数据加密方法针对的是数据安全和隐私保护这两个在商业领域中日益重要的技术问题,通过结合自动识别敏感数据、数据分区、与多方计算MPC任务的关联、动态增量加密等技术手段,实现敏感数据的自动识别与保护和数据隐私与安全性增强。

CN 117874789 A

1.一种动态的隐私数据加密方法,所述方法包括:

步骤S1:对敏感数据进行自动识别与分区;

其中,自动分析和标识敏感数据,并根据敏感性等级对其进行分区,每个分区的数据根据其与特定多方计算MPC任务的关联性进行数据分块,在每个分组形成多个数据块;

步骤S2:对数据块更新,并将数据块与MPC任务关联;

步骤S3:实行动态增量加密机制;

其中,相关的MPC参与者开启增量加密功能,以及基于数据块大小、当前任务处理情况以及资源可用情况对变更的数据实施增量加密;

其中,所述增量加密基于动态的权限密钥;

增量加密完成后,所述MPC参与者将增量加密后的数据分发给所有存储节点。

步骤S4:存储节点将增量加密后的数据应用到相关数据块中,同时保持未变更数据的加密状态不变。

2.如权利要求1所述的一种动态的隐私数据加密方法,其特征在于,所述步骤S2中,数据块更新,并将数据块与MPC任务关联,具体包括:

系统通过注册的事件监听器检测到这个更新;查找这个数据块关联的MPC任务,并确定当前任务受到影响;系统向参与当前MPC任务的参与者发送安全通知;MPC参与者接收到通知后,启动增量加密算法,只加密和传送更改的数据部分。

3.如权利要求1所述的一种动态的隐私数据加密方法,其特征在于,所述步骤S1中:

在开始自动标识之前,收集并预处理数据;

对预处理后的数据进行敏感性评估,包括进行关键词进行特征提取后,根据正则表达式进行关键词匹配,来识别商业数据中的敏感数据,基于预定义的敏感数据标识符和业务规则对预处理后的数据确定数据的敏感性等级;

根据敏感数据的等级,数据被分区为不同的类别。

4.如权利要求1所述的一种动态的隐私数据加密方法,其特征在于,所述步骤S1中:

数据分区完成,对数据分区划分为多个数据块,每一数据块与特定的MPC任务相关联;

MPC任务针对每个敏感性等级的数据块使用相应的加密协议和算法;

至少基于加密算法的计算复杂度、网络带宽、处理器能力共同来确定分区内数据块的组织方式。

5.如权利要求4所述的一种动态的隐私数据加密方法,其特征在于,初始化获取所有参数以计算数据块大小,包括如下参数:

C:加密算法计算复杂度;

P:处理器的处理能力;

B:网络带宽;

T_max:最大允许的加密操作延迟时间;

S_sec:安全性参数;

O:算法的开销;

然后,基于最大延迟计算数据块大小N1,包括基于每个加密操作的最大延迟时间,使用以下公式来确定数据块的大小:

$N1=(T\_max-O)*P/C$,

其中,

(T_max-O)表示实际用于数据加密的时间,P/C表示在单位时间内能够处理的数据量,用于限定CPU的处理能力上限对数据块大小的限制程度;

同时,基于网络带宽计算数据块大小N2,包括使用以下公式确定基于网络能够在T_max时间内传输的数据量来确定数据块的大小N2:N2＝B*T_max;其中,N2应该小于或等于网络在T_max内能够传输的最大数据量;

以及,根据安全性参数S_sec来根据不同的安全性要求来调整数据块的大小,对于需要更高安全性的数据,通过减小数据块的大小来增加安全性,用公式表示为:N_adj＝N*S_sec;

基于处理器能力、网络带宽、安全性要求来确定数据块的大小:

最终的数据块大小N_final是以上几个因素计算结果的最小值:

N_final＝min(N,N_adj,B*T_max),

所述N_final为对应分区内数据块的组织方式。

6.如权利要求5所述的一种动态的隐私数据加密方法,其特征在于,对于S_sec的确定,包括使用敏感性等级到安全性权重的映射确定。

7.如权利要求5所述的一种动态的隐私数据加密方法,其特征在于,对于S_sec的确定,包括根据以下公式计算S_sec:

S_sec＝W_sec*F_legal*F_threat*F_env,

其中,

W_sec:根据敏感性等级而定义的安全性权重;

F_legal:法律要求因子;

F_threat:威胁模型因子;

F_env:环境因子。

8.如权利要求5所述的一种动态的隐私数据加密方法,其特征在于,使用以下公式计算F_threat:

F_threat＝F_threat＝b+(b*(1-(Σ(Score_i)/(N*Max_Score)))),在此公式中:

b:设置的参数取值下限的常数,表明设置的因素影响的最大程度;

Score_i:第i个潜在威胁的评分;

N:潜在威胁的数量;

Max_Score:单个潜在威胁的最高评分。

9.如权利要求1所述的一种动态的隐私数据加密方法,其特征在于,所述步骤S3中实行动态增量加密机制,包括:系统根据所述数据块大小、当前的CPU负载、内存使用情况和网络带宽确定当前MPC任务的增量加密策略;

以及,基于当前数据版本号生成一个新的权限密钥,并基于权限密钥进行增量加密。

10.一种动态的隐私数据加密系统,包括存储器、处理器,其特征在于,所述存储器中存储计算机程序代码,所述处理器用于执行所述计算机程序代码实现权利要求1-9的动态的隐私数据加密方法。

一种动态的隐私数据加密方法和系统

技术领域

[0001] 本发明属于计算机技术领域,尤其涉及一种动态的隐私数据加密方法和系统。

背景技术

[0002] 随着信息技术的迅速发展,数据成为了企业中最宝贵的资源之一。尤其在大数据、云计算和物联网技术不断进步的背景下,数据的收集、存储、处理和分析变得更加频繁和复杂。商业数据通常包含了大量的敏感信息,如个人隐私数据、财务信息等,这些信息一旦泄露,可能会对企业造成重大的经济损失和信誉损害。

[0003] 目前,数据加密是保护商业数据不被未经授权访问的常用手段。然而,传统的数据加密方法通常在数据存储时将整个数据集进行加密,这种静态的加密方式存在一些缺陷。首先,当数据需要更新或处理时,可能需要对整个加密数据集进行解密和再加密,这不仅效率低下,而且在数据解密的过程中存在安全隐患。其次,静态加密方法不利于实现对数据访问权限的细粒度控制,不满足现代商业对数据安全性和灵活性的双重需求。

[0004] 因此,亟需一种新型的动态隐私数据加密方法和系统,该方法和系统能够实现对商业敏感数据的高效、安全管理,保护商业利益和客户信任。

发明内容

[0005] 针对上述现有技术中存在的缺陷,本发明提供一种动态的隐私数据加密方法,所述方法包括:

[0006] 步骤S1:对敏感数据进行自动识别与分区;

[0007] 其中,自动分析和标识敏感数据,并根据敏感性等级对其进行分区,每个分区的数据根据其与特定多方计算MPC任务的关联性进行数据分块,在每个分组形成多个数据块;

[0008] 步骤S2:对数据块更新,并将数据块与MPC任务关联;

[0009] 步骤S3:实行动态增量加密机制;

[0010] 其中,相关的MPC参与者开启增量加密功能,以及基于数据块大小、当前任务处理情况以及资源可用情况对变更的数据实施增量加密;

[0011] 其中,所述增量加密基于动态的权限密钥;

[0012] 增量加密完成后,所述MPC参与者将增量加密后的数据分发给所有存储节点。

[0013] 步骤S4:存储节点将增量加密后的数据应用到相关数据块中,同时保持未变更数据的加密状态不变。

[0014] 其中,所述步骤S2中,数据块更新,并将数据块与MPC任务关联,具体包括:

[0015] 系统通过注册的事件监听器检测到这个更新;查找这个数据块关联的MPC任务,并确定当前任务受到影响;系统向参与当前MPC任务的参与者发送安全通知;MPC参与者接收到通知后,启动增量加密算法,只加密和传送更改的数据部分。

[0016] 其中,所述步骤S1中:

[0017] 在开始自动标识之前,收集并预处理数据;

[0018] 对预处理后的数据进行敏感性评估,包括进行关键词进行特征提取后,根据正则表达式进行关键词匹配,来识别商业数据中的敏感数据,基于预定义的敏感数据标识符和业务规则对预处理后的数据确定数据的敏感性等级;

[0019] 根据敏感数据的等级,数据被分区为不同的类别。

[0020] 其中,所述步骤S1中:

[0021] 数据分区完成,对数据分区划分为多个数据块,每一数据块与特定的MPC任务相关联;

[0022] MPC任务针对每个敏感性等级的数据块使用相应的加密协议和算法;

[0023] 至少基于加密算法的计算复杂度、网络带宽、处理器能力共同来确定分区内数据块的组织方式。

[0024] 其中,初始化获取所有参数以计算数据块大小,包括如下参数:

[0025] C:加密算法计算复杂度;

[0026] P:处理器的处理能力;

[0027] B:网络带宽;

[0028] T_max:最大允许的加密操作延迟时间;

[0029] S_sec:安全性参数;

[0030] O:算法的开销;

[0031] 然后,基于最大延迟计算数据块大小N1,包括基于每个加密操作的最大延迟时间,使用以下公式来确定数据块的大小:

[0032] $N1=(T\_max-O)*P/C$,

[0033] 其中,

[0034] (T_max-O)表示实际用于数据加密的时间,P/C表示在单位时间内能够处理的数据量,用于限定CPU的处理能力上限对数据块大小的限制程度;

[0035] 同时,基于网络带宽计算数据块大小N2,包括使用以下公式确定基于网络能够在T_max时间内传输的数据量来确定数据块的大小N2:$N2=B*T\_max$;其中,N2应该小于或等于网络在T_max内能够传输的最大数据量;

[0036] 以及,根据安全性参数S_sec来根据不同的安全性要求来调整数据块的大小,对于需要更高安全性的数据,通过减小数据块的大小来增加安全性,用公式表示为:$N\_adj=N*S\_sec$;

[0037] 基于处理器能力、网络带宽、安全性要求来确定数据块的大小:

[0038] 最终的数据块大小N_final是以上几个因素计算结果的最小值:

[0039] $N\_final=\min(N,N\_adj,B*T\_max)$,

[0040] 所述N_final为对应分区内数据块的组织方式。

[0041] 其中,对于S_sec的确定,包括使用敏感性等级到安全性权重的映射确定。

[0042] 其中,对于S_sec的确定,包括根据以下公式计算S_sec:

[0043] $S\_sec=W\_sec*F\_legal*F\_threat*F\_env$,

[0044] 其中,

[0045] W_sec:根据敏感性等级而定义的安全性权重;

[0046] F_legal:法律要求因子;

[0047] F_threat:威胁模型因子;

[0048] F_env:环境因子。

[0049] 其中,使用以下公式计算F_threat:

[0050] F_threat＝F_threat＝b+(b*(1-(Σ(Score_i)/(N*Max_Score)))),在此公式中:

[0051] b:设置的参数取值下限的常数,表明设置的因素影响的最大程度;Score_i:第i个潜在威胁的评分;

[0052] N:潜在威胁的数量;

[0053] Max_Score:单个潜在威胁的最高评分。

[0054] 其中,所述步骤S3中实行动态增量加密机制,包括:系统根据所述数据块大小、当前的CPU负载、内存使用情况和网络带宽确定当前MPC任务的增量加密策略;

[0055] 以及,基于当前数据版本号生成一个新的权限密钥,并基于权限密钥进行增量加密。

[0056] 本发明还公开了一种动态的隐私数据加密系统,包括存储器、处理器,其特征在于,所述存储器中存储计算机程序代码,所述处理器用于执行所述计算机程序代码实现前述的动态的隐私数据加密方法。

[0057] 本发明自动识别敏感数据并对其进行分区,有助于减少人为错误,并提高敏感数据保护的效率和准确性。本发明通过与MPC任务关联并实施动态增量加密,只有被授权的参与者才能访问和处理数据,从而增强数据的安全性。此外,增量加密允许只对发生变更的数据块进行处理,还可以减少了计算量并提高了数据处理的灵活性和效率。

附图说明

[0058] 通过参考附图阅读下文的详细描述,本公开示例性实施方式的上述以及其他目的、特征和优点将变得易于理解。在附图中,以示例性而非限制性的方式示出了本公开的若干实施方式,并且相同或对应的标号表示相同或对应的部分,其中:

[0059] 图1是示出根据本发明实施例的一种动态的隐私数据加密方法的流程图。

具体实施方式

[0060] 为了使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明作进一步地详细描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其它实施例,都属于本发明保护的范围。

[0061] 在本发明实施例中使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本发明。在本发明实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义,“多种”一般包含至少两种。

[0062] 应当理解,尽管在本发明实施例中可能采用术语第一、第二、第三等来描述……,但这些……不应限于这些术语。这些术语仅用来将……区分开。例如,在不脱离本发明实施例范围的情况下,第一……也可以被称为第二……,类似地,第二……也可以被称为第一……。

[0063] 应当理解,本文中使用的术语“和/或”仅仅是一种描述关联对象的关联关系,表示

可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本文中字符“/”,一般表示前后关联对象是一种“或”的关系。

[0064] 取决于语境,如在此所使用的词语“如果”、“若”可以被解释成为“在……时”或“当……时”或“响应于确定”或“响应于检测”。类似地,取决于语境,短语“如果确定”或“如果检测(陈述的条件或事件)”可以被解释成为“当确定时”或“响应于确定”或“当检测(陈述的条件或事件)时”或“响应于检测(陈述的条件或事件)”。

[0065] 还需要说明的是,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的商品或者装置不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种商品或者装置所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的商品或者装置中还存在另外的相同要素。

[0066] 本发明提出的动态隐私数据加密方法主要针对的是数据安全和隐私保护这两个在商业领域中日益重要的技术问题。通过结合自动识别敏感数据、数据分区、与多方计算(MPC)任务的关联、动态增量加密等步骤,实现敏感数据的自动识别与保护和数据隐私与安全性增强。

[0067] 如图1所示,本发明公开了一种动态的隐私数据加密方法,所述方法包括:

[0068] 步骤S1:对敏感数据进行自动识别与分区。

[0069] 所述步骤S1中,需要自动分析和标识敏感数据,并根据敏感性等级对其进行分区。每个分区的数据根据其与特定MPC(多方计算)任务的关联性进行数据分块,在每个分组形成多个数据块。在每个分布式存储节点中应用同样的数据块组织结构,便于数据块结构的统一管理和存储,每个数据块内可以存储有不同版本的数据包。

[0070] 步骤S2:对数据块更新,并将数据块与MPC任务关联。

[0071] 所述步骤S2中,需要当某个数据块准备更新时,将其与相关的MPC任务联系起来。数据块发生变化时,通知相关MPC任务的所有参与者,激活增量加密机制。

[0072] 步骤S3:实行动态增量加密机制。

[0073] 所述步骤S3中,相关的MPC参与者开启增量加密功能。基于数据块大小、当前任务处理情况以及资源可用情况对变更的数据实施增量加密,其中,所述增量加密基于动态的权限密钥。增量加密完成后,所述MPC参与者将增量加密后的数据分发给所有存储节点。

[0074] 步骤S4:增量解密与应用结果

[0075] 所述步骤S4中,存储节点将增量加密后的数据应用到相关数据块中,同时保持未变更数据的加密状态不变。

[0076] 本发明自动识别敏感数据并对其进行分区,有助于减少人为错误,并提高敏感数据保护的效率和准确性。本发明通过与MPC任务关联并实施动态增量加密,只有被授权的参与者才能访问和处理数据,从而增强数据的安全性。此外,增量加密允许只对发生变更的数据块进行处理,还可以减少了计算量并提高了数据处理的灵活性和效率。

[0077] 为了实现自动识别和标记出敏感数据,并依据其敏感性等级将数据进行适当的分区。在开始自动标识之前,首先要收集并预处理数据。预处理可能包括清洗数据(移除重复项、修正错误、填补缺失值)、格式标准化(日期格式统一)以及文本标准化等。

[0078] 然后对预处理后的数据进行敏感性评估,包括进行关键词进行特征提取后,根据

正则表达式进行关键词匹配,来识别商业数据中的敏感数据。

[0079] 正则表达式为一种基于规则的算法,基于预定义的敏感数据标识符和业务规则对预处理后的数据确定数据的敏感性等级。

[0080] 的商业系统中,敏感数据包括但不限于个人身份信息(如身份证号码、手机号码)、财务信息(如银行卡号)、个人隐私(如住址、邮箱地址)等。其中,身份证号码(18位,最后一位可能是数字或字母X),手机号码(目前中国的手机号码为11位数字,且以13、14、15、16、17、18、19开头),银行卡号(通常为16到19位数字),邮箱地址(简单邮箱地址匹配),住址(匹配中文字符及常见地址组成部分)。随着规则和标准的更新,需要定期更新正则表达式以适应新的格式。

[0081] 根据敏感数据的等级,数据被分区为不同的类别。基于数据的敏感性等级(如等级1~5)来进行分区,其中低等级的数据敏感性更低,高等级的数据敏感性更高。

[0082] 对于敏感性等级,可选地,按照如下规则确定敏感性等级:

[0083] 敏感性等级1,例如公开发布的信息。敏感性等级2,例如客户个人邮箱地址。敏感性等级3,例如客户住址信息。敏感性等级4,例如客户手机号码、身份证号码。敏感性等级5,例如银行卡号、以及对应的银行账户交易信息。

[0084] 一旦数据分区完成,对数据分区划分为多个数据块,每一块数据需要与特定的MPC任务相关联。

[0085] MPC任务为对数据块数据执行加密处理,确保合适的加密强度。

[0086] 将数据块根据其敏感性等级与相应的MPC任务关联。确保MPC任务针对每个敏感性等级的数据块使用相应的加密协议和算法。基于加密算法的计算复杂度、网络带宽、处理器能力等因素共同来确定分区内数据块的组织方式。

[0087] 基于加密算法的计算复杂度划分数据块,包括当创建的数据块较小时,表明数据敏感性越高,需要应用计算复杂度高的加密算法。反之,当创建的数据块较大时,表明数据敏感性越低,可以应用计算复杂度低的加密算法,以减少加密操作的总次数。

[0088] 具体地,初始化获取所有参数以计算数据块大小,包括如下参数:

[0089] C:加密算法计算复杂度,用每比特需要的CPU周期数标识。

[0090] P:处理器的处理能力,用每秒可以处理的比特数(bps)表示。

[0091] B:网络带宽,以每秒可以传输的比特数(bps)表示。

[0092] T_max:最大允许的加密操作延迟时间(秒)。

[0093] S_sec:安全性参数,根据安全要求确定的数据块大小系数。

[0094] O:算法的开销,表示为非数据依赖的常数时间(秒),可以以算法初始化时间来确定非数据依赖算法开销。

[0095] 然后,基于最大延迟计算数据块大小N1,包括基于每个加密操作的最大延迟时间,使用以下公式来确定数据块的大小:

[0096] $N1 = (T\_max - O) * P / C$,

[0097] 其中,

[0098] (T_max-O)表示实际用于数据加密的时间,P/C表示在单位时间内能够处理的数据量,用于限定CPU的处理能力上限对数据块大小的限制程度。

[0099] 同时,基于网络带宽计算数据块大小N2,包括使用以下公式确定基于网络能够在

T_max时间内传输的数据量来确定数据块的大小N2:N2＝B*T_max。在这种情况下,N2应该小于或等于网络在T_max内能够传输的最大数据量。

[0100] 此外,还需要根据安全性参数S_sec来根据不同的安全性要求来调整数据块的大小。对于需要更高安全性的数据,通过减小数据块的大小来增加安全性,用公式表示为:N_adj＝N*S_sec。

[0101] 如果S_sec小于1,表示需要减小数据块的大小来提高安全性;如果S_sec大于1,则可以增加数据块的大小。

[0102] 最后,综合考虑处理能力、网络带宽、安全性要求等因素来确定数据块的大小。因此,最终的数据块大小N_final是以上几个因素计算结果的最小值:N_final＝min(N,N_adj,B*T_max)。

[0103] 所述N_final为对应分区内数据块的组织方式,由于不同分区的S_sec不相同,且分区对应分配的CPU的处理能力P可能不相同,不同分区的加密算法复杂度可能也不相同,因此不同的分区内的数据块的组织方式不相同。

[0104] 其中,对于S_sec的确定,有以下两种确定方式:

[0105] 方式一、使用敏感性等级到安全性权重的映射。

[0106] 具体为,为每个等级分配一个安全性权重W_sec,它根据数据的敏感性等级来缩放。

[0107] 等级1:W_sec＝1.0(默认权重,不需要额外的安全措施)

[0108] 等级2:W_sec＝0.8

[0109] 等级3:W_sec＝0.6

[0110] 等级4:W_sec＝0.4

[0111] 等级5:W_sec＝0.2(最高等级的敏感性,需要最小的数据块)

[0112] 这里,W_sec值小于1表明随着敏感性等级的升高,需要减小数据块的大小来提高安全性。

[0113] 方式二、如果数据分区只是基于数据属性来进行,在一个分区内有多个不同属性的数据内容,对数据分块时,保证一个数据块内只有同一属性的数据内容。例如,如果等级1的分区中,同时有银行卡号、交易记录、交易总额,交易时间等等信息,但对于某一具体的数据块,只保存银行卡号,则可能根据方式二确定同一数据分区内不同数据块的安全性权重W_sec,使得安全性权重W_sec的分配更加灵活。

[0114] 以下为方式二中计算S_sec的公式:

[0115] S_sec＝W_sec*F_legal*F_threat*F_env,

[0116] 其中,

[0117] W_sec:根据敏感性等级而定义的安全性权重(参照方式一)。

[0118] F_legal:法律要求因子,如果有法律要求的严格程度确定F_legal,这个值在常数b(例如0.5)到1之间。

[0119] F_threat:威胁模型因子,根据高风险的威胁成都确定F_threat,这个值在常数b(例如0.5)到1之间。

[0120] F_env:环境因子,根据数据处理环境的安全程度确定,这个值在常数b(例如0.5)到1之间。

[0121] 法律要求因子(F_legal)反映了组织必须遵守的数据保护和隐私法律的严格程度。这个因素可以基于法律要求的复杂性、罚款的严重性以及合规成本来确定。其中,F_legal的值应当在0到1之间,值越小表示法律要求越严格,需要更多的安全措施以确保合规。

[0122] 威胁模型因子(F_threat)考虑了组织面临的安全威胁的类型和严重性。不同属性类型的数据和不同的业务模型面临的威胁模型会有所不同,例如银行卡信息可能相对于邮箱信息面临更高的威胁。可选地,F_threat的值在常数b(例如0.5)到1之间,值越大表示潜在威胁的可能性和影响越高,需要更多的安全措施以减轻风险。

[0123] 可以基于评分计算的方式来确定F_threat,具体包括:

[0124] 识别可能影响数据安全的所有潜在威胁。例如:外部攻击(如DDoS攻击、钓鱼攻击);内部威胁(如恶意内部人员、误操作);系统漏洞(如软件未更新)。

[0125] 评估威胁的可能性和影响,具体对于每个潜在威胁,评估其发生的可能性和对组织产生的影响。可以使用以下评分:

[0126] 0分:威胁不可能发生或影响微不足道。

[0127] 1分:威胁可能性低,影响较小。

[0128] 2分:威胁可能性中等,影响显著。

[0129] 3分:威胁可能性高,影响严重。

[0130] 最后,计算F_threat,使用以下公式计算F_threat:

[0131] $F\_threat = F\_threat = b + (b * (1 - (\Sigma (Score\_i) / (N * Max\_Score))))$,

[0132] 在此公式中:

[0133] b:设置的参数取值下限的常数,表明设置的因素影响的最大程度,例如设置为0.5。

[0134] Score_i:第i个潜在威胁的评分。

[0135] N:潜在威胁的数量。

[0136] Max_Score:单个潜在威胁的最高评分(本例中为3)。

[0137] 基于上述公式,在所有潜在威胁的评分都是最高的情况下(即最不安全),F_threat的值将为b,而如果所有潜在威胁的评分都是最低的情况下(即最安全),F_threat的值将是1(b为0.5的情形下)。

[0138] 对于环境因子F_env,可以根据与数据处理环境安全相关的控制措施,包括物理安全措施(如内网、外网),以及网络安全设备(如防火墙、入侵检测系统)的配置程度来确定安全程度,安全程度越高的环境环境因子F_env越高,安全程度越低的环境环境因子F_env越低。

[0139] 对于步骤S2中对于数据块更新,并将数据块与MPC任务关联,包括:系统通过注册的事件监听器检测到这个更新;查找这个数据块关联的MPC任务,并确定当前任务受到影响;系统向参与当前MPC任务的参与者发送安全通知;MPC参与者接收到通知后,启动增量加密算法,只加密和传送更改的数据部分;MPC参与者更新本地的计算,重新计算或调整自身负责的部分。

[0140] 首先,必须有一个机制来监测数据块何时准备更新,包括通过版本控制、触发器或事件监听来实现。数据所有者或控制者应对数据块进行版本管理,并在数据块将要被更新

时生成更新事件。加一段版本升级的触发条件,可以是数据积累量决定。

[0141] 每个数据块更新事件需要与一个或多个MPC任务相关联,包括维护一个映射,记录哪些MPC任务依赖于哪些数据块,当数据块更新时,查询这个映射找出受影响的MPC任务。

[0142] 当更新发生后,系统需要通知相关MPC任务的参与者。这可以通过一个安全的消息传递系统来实现,确保参与者收到通知,并了解所依赖的数据块已更新。

[0143] 由于MPC任务通常涉及敏感数据,因此在更新数据块时应使用增量加密机制来保护数据的安全性,在此过程中,只有变化的部分被重新加密和分发,而不是整个数据块。

[0144] 参与者在收到更新通知后,需要根据新的数据块内容调整自身计算部分,包括重新运行某些计算步骤或完全重新开始MPC过程。

[0145] 对于一个MPC实例,每个MPC参与者可以基于以下框架,包括使用开源的MPC协议库,通过TLS(传输层安全)或DTLS(数据报传输层安全)协议来保护数据在网络上传输,利用分布式锁或共识算法来实现操作的同步,使用数据库加密、安全存储或者密钥管理系统来管理数据,部署HSM以增强密钥的安全性和加密操作,以及在支持SGX的CPU上创建安全飞地来执行敏感计算。

[0146] 所述步骤S3中实行动态增量加密机制,包括:系统根据所述数据块大小、当前的CPU负载、内存使用情况和网络带宽确定当前MPC任务的增量加密策略。以及,系统基于当前数据版本号生成一个新的权限密钥,并基于权限密钥进行增量加密。该步骤中仅对已更改的数据块应用加密,而不重新加密整个数据集。加密的增量版本的数据块通过安全的通信渠道发送到所有的存储节点。每个存储节点确认收到的数据块,并对其进行验证和同步,管理各个存储节点的数据的版本,确保所有参与者都在使用最新的加密数据块。

[0147] 实现对不同版本数据块的精细化管理,以及并行加密,提高加密速度。

[0148] 可选地,权限密钥的生成可以基于当前的数据版本号,结合一个密钥生成函数KeyGen,能使用散列算法或伪随机函数来从数据版本号生成密钥:

[0149] Permission_Key=KeyGen(Data_Version),

[0150] 这里的KeyGen函数是一个确定性函数,以确保相同的输入(数据版本号)将产生相同的权限密钥。确定性函数KeyGen用于生成权限密钥,其核心特性是对于相同的输入总是产生相同的输出。KeyGen通常会借助密码学安全的哈希函数或伪随机函数,输入数据版本号来生成密钥,例如Permission_Key=SHA256(Data_Version),其中Data_Version是密钥生成的主要输入,SHA256是一个单向的哈希函数,保证无法逆向推导出原始输入。由于KeyGen使用SHA256哈希函数来生成密钥,Permission_Key的长度将是固定的,因为SHA256的输出长度是固定的。SHA256哈希函数的输出长度是256位(即32字节)。无论输入数据的长度如何,SHA256始终产生一个固定长度的输出。这意味着Permission_Key的长度在使用SHA256时将始终是256位。

[0151] 可选地,所述数据块大小、当前的CPU负载、内存使用情况和网络带宽确定当前MPC任务的增量加密策略,使得根据系统的性能参数动态调整加密操作的参数,包括加密算法的选择、密钥的大小、加密操作的并发度。

[0152] 以下是增量加密策略制定的计算公式和参数定义:

[0153] BlockSize_Changed:待增量加密的数据包所在数据块的大小,BlockSize_Changed的数值为增量加密所在数据包对应的数据块在步骤S1中确定的N_final。

[0154] CPU_Load:当前CPU负载的百分比,表示为0到100之间的数值。

[0155] Memory_Usage:当前使用的内存量。

[0156] Memory_Total:系统总内存量。

[0157] Network_Bandwidth:当前可用的网络带宽。

[0158] Data_Version:当前数据的版本号。

[0159] 根据系统资源和数据块状态,制定以下增量加密策略:

[0160] (1)密钥长度Key_Length:

[0161] Key_Length=f(CPU_Load,Memory_Usage,Network_Bandwidth)

[0162] 其中f函数可以根据系统资源参数选择合适的密钥长度。高CPU负载和内存使用可能意味着选择一个较短的密钥长度来减少计算开销。

[0163] 可选地,函数f基于系统资源参数选择密钥长度,密钥长度的选择需要在安全性和性能之间做出平衡。可选地,根据CPU负载、内存使用和网络带宽来动态调整密钥长度。

[0164] 定义密钥长度的基准,分别为:

[0165] Key_Length_Max:可选的最大密钥长度,例如256位。

[0166] Key_Length_Min:可选的最小密钥长度,例如128位。

[0167] 定义资源使用的阈值,包括:

[0168] CPU_Load_Max:CPU负载的最大可接受百分比,例如80%。

[0169] Memory_Usage_Max:内存使用的最大可接受百分比,例如75%。

[0170] Network_Bandwidth_Min:网络带宽的最小可接受值,例如1Mbps。

[0171] 确定Key_Length=f(CPU_Load,Memory_Usage,Network_Bandwidth),即根据CPU负载、内存使用和网络带宽来动态调整密钥长度Key_Length。

[0172] 首先,将内存使用Memory_Usage转换为百分比Memory_Usage_Percent:Memory_Usage_Percent=(Memory_Usage/Memory_Total)*100

[0173] 然后,计算资源压力指数Stress_Index,值域为[0,1]:

[0174] Stress_Index=(CPU_Load/CPU_Load_Max+Memory_Usage_Percent/Memory_Usage_Max+(Network_Bandwidth_Min/Network_Bandwidth))/3,

[0175] 根据资源压力指数Stress_Index选择密钥长度Key_Length_1:

[0176] Key_Length_1=Key_Length_Max-(Stress_Index*(Key_Length_Max-Key_Length_Min)),

[0177] 最后,需要确保密钥Key_Length长度不低于最小值:

[0178] Key_Length=max(Key_Length_1,Key_Length_Min)。

[0179] 其中,Stress_Index是一个计算指标,综合考虑了CPU负载、内存使用百分比和网络带宽。当Stress_Index值增加时,表示系统资源的压力增大。密钥长度Key_Length是在最大值Key_Length_Max和最小值Key_Length_Min之间动态调整的,当系统压力增大时,选择较短的密钥长度以减少计算开销。上述f函数确保了密钥长度不会低于定义的最小安全标准Key_Length_Min。

[0180] 其中,Key_Length变量定义加密密钥的长度,Key_Length用于控制Permission_Key的密钥长度,系统中的加密密钥长度(Key_Length)需要动态调整为小于等于SHA256输出的长度,需要对SHA256的输出进行截断操作,以匹配所需的Key_Length。具体地,对对

SHA256的输出的Permission_Key截取前Key_Length位,得到一个长度为Key_Length的密钥。

[0181] (2)加密算法Encryption_Algorithm的选择:

[0182] Encryption_Algorithm=g(BlockSize_Changed),

[0183] 其中g函数可以根据待增量加密的数据包所在数据块的大小选择合适的加密算法。由于待增量加密的数据包所在数据块的大小体现了数据包内数据的安全敏感程度,因此较小的数据块需要使用计算量更大的加密算法,以提供更高的安全性。

[0184] 可选地,g函数应该基于待增量加密的数据包所在数据块的大小来选择不同的加密算法,定义对待增量加密的数据包所在数据块的大小划分区间的两个阈值:BlockSize_Small_Threshold:定义小数据块的最大阈值,BlockSize_Medium_Threshold:定义中等数据块的最大阈值。

[0185] 根据待增量加密的数据包所在数据块的大小所在不同区间确定对应的加密算法,包括:

[0186] 在待增量加密的数据包所在数据块的大小小于BlockSize_Small_Threshold时,进入Algorithm_Secure模式,采用更安全的加密算法RSA非对称加密算法。

[0187] 在待增量加密的数据包所在数据块的大小在BlockSize_Small_Threshold和BlockSize_Medium_Threshold之间时,进入Algorithm_Balanced模式,采用安全性和性能平衡的加密算法AES-CBC算法。

[0188] 在待增量加密的数据包所在数据块的大小大于BlockSize_Medium_Threshold时,进入Algorithm_Fast模式,采用计算量较小,执行更快的加密算法AES-CTR算法。

[0189] (3)加密操作的并发度Encryption_Concurrency:

[0190] Encryption_Concurrency=h(Network_Bandwidth,Memory_Usage)

[0191] 其中h函数可以根据可用网络带宽和内存使用来决定并发执行加密任务的数量。网络带宽较大且内存使用不高时,可以增加并发度来加速加密过程。

[0192] 根据网络带宽和内存使用来确定可以同时执行的加密任务数量的h函数,包括:

[0193] 定义资源使用的阈值:

[0194] Network_Bandwidth_Max:网络带宽的最大值。

[0195] Memory_Usage_Max:系统总内存中可以分配给加密操作的最大内存使用量。

[0196] 确定定义并发度的范围:

[0197] Concurrency_Min:最小并发度,例如1(表示至少有一个加密操作在运行)。

[0198] Concurrency_Max:最大并发度,例如256(基于系统能够处理的最大并发数)。

[0199] h函数的具体形式为:h(Network_Bandwidth,Memory_Usage),包括以下过程:

[0200] 首先,计算当前网络带宽利用率,假设更高的带宽利用率允许更多的并发加密操作:

[0201] Network_Utilization=Network_Bandwidth/Network_Bandwidth_Max,

[0202] 然后,计算当前内存利用率,更低的内存利用率允许更多的并发加密操作,则有:

[0203] Memory_Utilization=Memory_Usage/Memory_Usage_Max,Memory_Available=1-Memory_Utilization。

[0204] 之后,计算加密操作的并发度Encryption_Concurrency_1,取决于网络资源利用

率Network_Utilization和内存资源的利用率Memory_Available：

[0205] Encryption_Concurrency_1＝int((Network_Utilization+Memory_Available)/2*Concurrency_Max)。

[0206] 最后,对Encryption_Concurrency_1的范围进行限定处理,得到最终的加密操作的并发度Encryption_Concurrency,包括：

[0207] Encryption_Concurrency＝max(Encryption_Concurrency_1,Concurrency_Min),

[0208] Encryption_Concurrency＝min(Encryption_Concurrency_1,Concurrency_Max)。

[0209] 其中,Network_Utilization和Memory_Available是两个介于0和1之间的因子,分别代表网络带宽和可用内存的利用率。

[0210] 通过取网络带宽利用率和内存可用率的平均值,计算出一个标准化的并发度因子,并将其乘以Concurrency_Max来得到初始的并发度数目。

[0211] 然后,通过确保并发度不低于Concurrency_Min且不高于Concurrency_Max来调整最终的并发度。int函数用于确保并发度是一个整数,在数学上等于往下取整的计算函数。

[0212] 最后,基于确定的动态增量加密策略应用加密：

[0213] Encrypted_Block＝Encrypt(Data,Permission_Key,Encryption_Algorithm)

[0214] 其中,Encrypt是加密函数,Data是待加密的增量数据包,Permission_Key是用于加密的权限密钥,而Encryption_Algorithm是根据策略选择的算法。

[0215] 系统可以动态调整其增量加密方式,以最佳方式利用当前的系统资源,同时保证数据的安全性。

[0216] Encryption_Algorithm是加密算法,它定义了如何使用限制长度的密钥Permission_Key和选定加密算法在设定并发度的条件下来加密Data。所述设定算法可能是RSA,AES-CTR以及AES-CBC算法。所述设定并发度区间为1-256,所述限制长度为128-256位。

[0217] 最终,Encrypt函数结合了Data、Permission_Key和Encryption_Algorithm来生成Encrypted_Block。

[0218] 可选地,在多方计算(MPC)场景中,参与者通常需要共享信息,同时保持数据的隐私性。MPC参与者在完成增量加密后,会将加密后的数据块(Encrypted_Block)分发给所有的存储节点。

[0219] MPC参与者首先确定自上次更新以来哪些数据发生了变化。然后,他们使用Permission_Key和选定的Encryption_Algorithm对这些增量数据进行加密,生成Encrypted_Block。

[0220] 加密完成后,MPC参与者将Encrypted_Block发送到网络中的每一个存储节点。

[0221] 所述MPC参与者将限制密钥长度Key_Length和Encrypted_Block一并发送到网络中的每一个存储节点。

[0222] 存储节点收到Encrypted_Block后,会将其存储在本地。由于数据是加密的,即使存储节点被攻击或受到破坏,数据内容也能保持安全。

[0223] 在分布式存储系统中,用户通常需要通过安全的方式访问存储在节点上的加密数据。本发明中,当需要访问或处理这些数据时,授权的用户可以基于相应的解密密钥存储节

点获得解密的Encrypted_Block,从而获取原始的增量数据,包括:

[0224] 用户向存储节点发起请求前,需要通过系统的身份验证,确认用户是否具有访问特定版本号对应数据的权限。一旦验证通过,用户请求特定版本号的数据包及其对应的密钥长度Key_Length。存储节点接收到请求后,验证用户的请求是否有效后,所述验证用户请求有效,包括确认是否在验证通过后预设时间段内发送所述请求。一旦确认用户有权访问数据,存储节点将请求的数据包(仍然是加密状态)及所需的Key_Length发送给用户。

[0225] 用户根据提供的版本号和Key_Length,通过密钥生成机制(KeyGen函数)生成对应的加密密钥。版本号和密钥长度是公开的,密钥本身是私有的。

[0226] 用户使用生成的密钥对接收到的加密数据进行解密。解密成功后,用户可以访问数据的明文版本并进行后续操作,如阅读、编辑或处理数据。

[0227] 通过本发明的方法和系统,使用增量加密和解密减少了对整个数据集的加密需求,仅针对变化的部分进行处理,从而提高了数据处理的速度。此外,通过动态的权限密钥和与MPC任务关联的数据管理机制,允许对数据访问权限进行细粒度控制,提高了权限管理的灵活性。以及,通过对敏感数据进行分区并实施增量加密,本发明极大地强化了数据隐私保护,降低了数据泄露的风险,并通过动态增量加密根据当前任务处理情况和资源可用情况来进行,有助于优化资源使用,降低运维成本。

[0228] 需要说明的是,本公开上述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本公开中,计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读信号介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:电线、光缆、RF(射频)等等,或者上述的任意合适的组合。

[0229] 上述计算机可读介质可以是上述电子设备中所包含的;也可以是单独存在,而未装配入该电子设备中。

[0230] 可以以一种或多种程序设计语言或其组合来编写用于执行本公开的操作的计算机程序代码,上述程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(LAN)

或广域网(WAN)—连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0231] 附图中的流程图和框图,图示了按照本公开各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,该模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的是,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0232] 描述于本公开实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现。其中,单元的名称在某种情况下并不构成对该单元本身的限定。

[0233] 以上介绍了本发明的较佳实施方式,旨在使得本发明的精神更加清楚和便于理解,并不是为了限制本发明,凡在本发明的精神和原则之内,所做的修改、替换、改进,均应包含在本发明所附的权利要求概括的保护范围之内。


步骤S1：对敏感数据进行自动识别与分区
步骤S2：对数据块更新，并将数据块与MPC任务关联
步骤S3：实行动态增量加密机制
步骤S4：存储节点将增量加密后的数据应用到相关数据块中，同时保持未变更数据的加密状态不变

图1