(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111512589 A
(43)申请公布日 2020.08.07

(21)申请号 201880080281.2

(22)申请日 2018.12.13

(30)优先权数据
62/598666 2017.12.14 US

(85)PCT国际申请进入国家阶段日
2020.06.12

(86)PCT国际申请的申请数据
PCT/EP2018/084769 2018.12.13

(87)PCT国际申请的公布数据
WO2019/115697 EN 2019.06.20

(71)申请人 罗伯特·博世有限公司
地址 德国斯图加特

(72)发明人 范欣欣 郑清吉 J·瓜贾多默钱

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 张凌苗 申屠伟进

(51)Int.Cl.
H04L 9/00(2006.01)
H04L 9/08(2006.01)
H04L 9/32(2006.01)

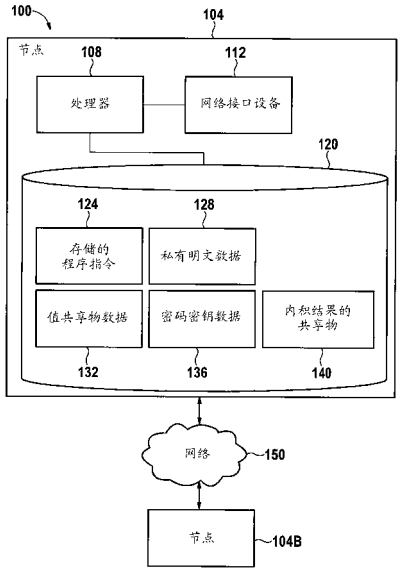
权利要求书2页 说明书11页 附图12页

(54)发明名称

用于利用SPDZ的快速安全多方内积的方法

(57)摘要

一种用于使用SPDZ协议在两个当事方之间实现安全多方内积计算的方法，牵涉到：对于$i = 1, \ldots, k$，使第一当事方和第二当事方分别基于向量($\mathbf{x} = \{x_1, \ldots, x_N\}$)和向量($\mathbf{w} = \{W_1, W_N\}$)来计算向量(I)=(II)，其中(I) = $(X_{2i-1}X_{2i})$，(III) = $W_{2i-1}W_{2i}$，N是向量中的元素总数，$k = N/2$。向量(I)和(III)在当事方之间安全地共享。当事方然后联合计算SPDZ协议$\mathrm{Add}([w_{2i}], [x_{2i-1}])$和$\mathrm{Add}([w_{2i}], [x_{2i-1}])$来分别确定共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$，并且然后对于$i = 1, \ldots, k$，通过执行SPDZ协议$\mathrm{Mult}([w_{2i-1} + x_{2i}], [w_{2i} + x_{2i-1}])$来计算内积共享物$[d_i]$。然后执行SPDZ协议([Add d_1],...,[d_k], -(IV),...,-(V), -(VI), -, (VII))来确定内积。



CN 111512589 A

1.一种用于实现安全多方内积计算协议的方法,其中第一计算当事方持有第一私有向量($x = \{x_1, \ldots, x_N\}$)并且第二计算当事方持有第二私有向量($w = \{w_1, \ldots, w_N\}$),第一私有向量和第二私有向量各自具有数量为N的值,其中第一计算当事方和第二计算当事方想要在第一计算当事方不学习第二私有向量并且第二计算当事方不学习第一私有向量的情况下,联合计算第一私有向量和第二私有向量的内积(d),使得$x_1 w_1 + \cdots + x_N w_N = \sum_{i=1}^{N} x_i w_i$,所述方法包括:

使用SPDZ协议将具有a的第一私有向量从第一计算当事方安全地共享给第二计算当事方;

对于$i = 1, \ldots, k$,为第一计算当事方计算向量$\overline{x_i} = \{\overline{x_1}, \ldots, \overline{x_k}\}$,其中$k = N/2$且值$\overline{x_i} = x_{2i-1} x_{2i}$,以及使用SPDZ协议与第二计算当事方安全地共享向量$\overline{x_i}$;

使用SPDZ协议将第二私有向量从第一计算当事方安全地共享给第二计算当事方;

对于$i = 1, \ldots, k$,为第二计算当事方计算向量$\overline{w_i} = \{\overline{w_1}, \ldots, \overline{w_k}\}$,其中值$\overline{w_i} = w_{2i-1} w_{2i}$,以及使用SPDZ协议与第一计算当事方安全地共享向量$\overline{w_i}$;

对于$i = 1, \ldots, k$,通过在第一计算当事方和第二计算当事方的情况下分别执行SPDZ协议$\mathrm{Add}([w_{2i}], [x_{2i-1}])$和$\mathrm{Add}([w_{2i}], [x_{2i-1}])$来计算共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$,共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$分别是$(w_{2i-1} + x_{2i})$和$(w_{2i} + x_{2i-1})$的共享物;

对于$i = 1, \ldots, k$,通过在第一计算当事方和第二计算当事方的情况下执行SPDZ协议$\mathrm{Mult}([w_{2i-1} + x_{2i}], [w_{2i} + x_{2i-1}])$来计算内积共享物$[d_i]$;以及

通过在第一计算当事方和第二计算当事方的情况下执行SPDZ协议$\mathrm{Add}([d_1], \ldots, [d_k], -[\overline{x_1}], \ldots, -[\overline{x_k}], -[\overline{w_1}], \ldots, -[\overline{w_k}])$来计算内积($d$)。

2.根据权利要求1所述的方法,其中第一私有向量是由第一计算当事方拥有的数据,并且第二私有向量是权重向量。

3.根据权利要求2所述的方法,其中第二计算当事方是机器学习服务提供者。

4.根据权利要求1所述的方法,其中SPDZ共享协议是加性秘密共享协议。

5.根据权利要求4所述的方法,其中,第一计算当事方具有消息认证码(MAC)密钥的第一MAC密钥共享物,并且第二计算当事方具有所述MAC密钥的第二MAC密钥共享物,使得第一MAC密钥共享物和第二MAC密钥共享物相加在一起得到所述MAC密钥。

6.根据权利要求5所述的方法,其中作为SPDZ共享协议的部分,第一MAC密钥由第一计算当事方共享给第二计算当事方,并且第二MAC密钥由第二计算当事方共享给第一计算当事方。

7.根据权利要求1所述的方法,其中计算共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$、内积共享物$[d_i]$以及内积(d)发生在SPDZ协议的在线阶段期间。

8.根据权利要求7所述的方法,其中第一计算当事方和第二计算当事方二者均在SPDZ协议的离线阶段期间生成Beaver三元组,所述Beaver三元组由第一计算当事方和第二计算当事方在所述在线阶段期间使用。

9.一种多方计算系统,包括:

具有第一私有向量($x = \{x_1, \ldots, x_N\}$)的第一计算当事方,其中N是预定数量;

具有第二向量($w = \{w_1, \ldots, w_N\}$)的第二计算当事方;以及

至少一个安全通信信道，经由所述至少一个安全通信信道，第一计算当事方和第二计算当事方可以安全地向彼此传输消息；

其中第一计算当事方和第二计算当事方被配置为基于SPDZ协议执行安全多方内积计算，以在第一计算当事方不学习第二私有向量并且第二计算当事方不学习第一私有向量的情况下，联合计算第一私有向量和第二私有向量的内积(*d*)，使得$x_1w_1 + \cdots + x_Nw_N = \sum_{i=1}^{N}$，

其中安全多方内积计算包括：

使用SPDZ协议将具有*a*的第一私有向量从第一计算当事方安全地共享给第二计算当事方，

对于$i = 1, \ldots, k$，为第一计算当事方计算向量$\overline{x_i} = \{\overline{x_1}, \ldots, \overline{x_k}\}$，其中$k = N/2$且值$\overline{x_i} = x_{2i-1}x_{2i}$，以及使用SPDZ协议与第二计算当事方安全地共享向量$\overline{x_i}$，

使用SPDZ协议将第二私有向量从第一计算当事方安全地共享给第二计算当事方，

对于$i = 1, \ldots, k$，为第二计算当事方计算向量$\overline{w_i} = \{\overline{w_1}, \ldots, \overline{w_k}\}$，其中值$\overline{w_i} = w_{2i-1}w_{2i}$，以及使用SPDZ协议与第一计算当事方安全地共享向量$\overline{w_i}$，

对于$i = 1, \ldots, k$，通过在第一计算当事方和第二计算当事方的情况下分别执行SPDZ协议$\mathrm{Add}([w_{2i}], [x_{2i-1}])$和$\mathrm{Add}([w_{2i}], [x_{2i-1}])$来计算共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$，共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$分别是$(w_{2i-1} + x_{2i})$和$(w_{2i} + x_{2i-1})$的共享物，

对于$i = 1, \ldots, k$，通过在第一计算当事方和第二计算当事方的情况下执行SPDZ协议$\mathrm{Mult}([w_{2i-1} + x_{2i}], [w_{2i} + x_{2i-1}])$ 来计算内积共享物$[d_i]$，以及

通过在第一计算当事方和第二计算当事方的情况下执行SPDZ协议$\mathrm{Add}([d_1], \ldots, [d_k], -[\overline{x_1}], \ldots, -[\overline{x_k}], -[\overline{w_1}], \ldots, -[\overline{w_k}])$来计算内积(*d*)。

10.根据权利要求9所述的系统，其中第一私有向量是由第一计算当事方拥有的数据，并且第二私有向量是权重向量。

11.根据权利要求10所述的系统，其中第二计算当事方是机器学习服务提供者。

12.根据权利要求9所述的系统，其中SPDZ共享协议是加性秘密共享协议。

13.根据权利要求12所述的系统，其中，第一计算当事方具有消息认证码(MAC)密钥的第一MAC密钥共享物，并且第二计算当事方具有所述MAC密钥的第二MAC密钥共享物，使得第一MAC密钥共享物和第二MAC密钥共享物相加在一起得到所述MAC密钥。

14.根据权利要求13所述的系统，其中作为SPDZ共享协议的部分，第一MAC密钥由第一计算当事方共享给第二计算当事方，并且第二MAC密钥由第二计算当事方共享给第一计算当事方。

15.根据权利要求9所述的系统，其中计算共享物$[w_{2i-1} + x_{2i}]$和$[w_{2i} + x_{2i-1}]$、内积共享物$[d_i]$以及内积(*d*)发生在SPDZ协议的在线阶段期间。

16.根据权利要求15所述的系统，其中第一计算当事方和第二计算当事方二者均在SPDZ协议的离线阶段期间生成Beaver三元组，所述Beaver三元组由第一计算当事方和第二计算当事方在所述在线阶段期间使用。

用于利用SPDZ的快速安全多方内积的方法

[0001] 相关申请的交叉引用

本申请要求Zheng等人于2017年12月14日提交的题为“METHOD FOR FASTER SECURE MULTIPARTY INNER PRODUCT WITH SPDZ”的美国临时申请序列号62/598,666的优先权,特此将该美国临时申请的公开内容通过引用在其整体上并入本文中。

技术领域

[0002] 本公开总地涉及多方计算(MPC)协议,并且特别地涉及SPDZ(发音为“speedz”)MPC协议。

背景技术

[0003] 在安全多方计算(MPC)协议中,多个当事方关于其私有输入联合计算函数并且仅学习函数的输出。SPDZ协议是最实用和活跃的安全MPC协议之一,并且实现针对任何数量的恶意受损方的安全性。具体来说,SPDZ协议正在预处理模型中工作:在不知道任何未来输入的情况下,离线阶段执行耗时的任务来生成元组;并且在线阶段利用私有输入以及运用来自离线阶段的那些元组来执行特定函数。特别地,SPDZ是允许运算电路的联合计算的通用MPC协议。

[0004] 安全内积是隐私保护数据挖掘中广泛使用的构造块。如本领域公知的,内积运算(也称为“点积”)将来自两个向量的对应元素对乘在一起以产生个体乘积,并且将所有乘积求和以产生最终的标量总和值。例如,如果w =(1,2,3)并且x = (4,5,6),那么内积是:$(1\cdot 4)+(2\cdot 5)+(3\cdot 6)=32$。内积运算用于宽范围的实际计算中,并且举一个非限制性的示例,神经网络大量使用内积运算,其中向量w表示在神经网络中使用的权重向量,并且向量x表示输入值,其中神经元运算的部分是为了生成每个权重乘以对应输入值的总和,其是上述内积运算。SPDZ协议使得各自具有对仅输入数据的部分的访问的两个或更多个当事方使用安全多方计算来生成内积值。例如,在一种配置中,一个计算设备具有对神经网络中的权重向量w的访问,并且另一个计算设备具有对输入向量x的访问,该输入向量x可以包含关于输入图像或隐私敏感的其他输入数据的信息。SPDZ协议使得能够计算内积的最终输出,而不向存储权重值w的当事方透露输入向量x并且不向存储输入向量x的当事方透露权重值w。SPDZ协议在至少两个当事方的群组之间使能实现该运算,尽管在一些配置中,向量w和x在更大的当事方群组间被进一步细分。虽然本文中没有更详细地描述,但是SPDZ协议还为群组中的当事方提供了用以确定当事方的子集是否已经操纵了共享数据来生成不正确的最终结果的某种能力,这为群组提供了用以确定最终的计算结果是否已经被准确地计算或者应当被丢弃的某种能力。

[0005] 对于两个N维向量$x=(x_1,x_2,\ldots,x_N)$和$w=(w_1,w_2,\ldots,w_N)$,它们的内积是总和$\sum_{i=1}^{N}x_iw_i$。利用SPDZ对安全内积$x\cdot w$的直接实现牵涉到在线阶段中的N个逐对乘法、继之以$N-1$个加法。找到用于降低针对安全内积的计算要求的方式可以改进针对SPDZ的在线阶段和离线阶段二者的性能。

附图说明

[0006] 图1描绘了SPDZ协议$\mathbf{Share}(\mathrm{ct}_s, P_1, \ldots, P_n)$的步骤，其中$\mathrm{ct}_s = \mathrm{Enc}_{\mathrm{pk}}(s)$是公共已知的，并且$P_1, \ldots, P_n$是$n$个当事方。

[0007] 图2描绘了SPDZ协议$\mathbf{Add}([x^1], [x^2], \ldots, [x^l])$的步骤，其中$[x^j]$是共享值$x^j (1 \le j \le l)$。

[0008] 图3描绘了SPDZ协议$\mathbf{Mult}([x], [y])$的步骤，其中[x]是值x的共享物，并且[y]是值y的共享物。

[0009] 图4是被配置成用改进的效率执行内积的安全多方计算的系统的示意图。

[0010] 图5描绘了根据本公开的用于容错SPDZ的协议ReShare的算法。

[0011] 图6描绘了用于容错SPDZ的协议ReCombine的算法。

[0012] 图7描绘了在一当事方将其持有的秘密共享物重新分配给其他当事方之后的中间结果的表格。

[0013] 图8描绘了示出在每个当事方将其秘密共享物重新分配给其他当事方之后的最终结果的表格。

[0014] 图9描绘了示出在执行协议ReShare之后的最终结果的表格。

[0015] 图10描绘了用于基于协议ReShare的秘密共享物重新分配阶段的算法。

[0016] 图11描绘了用于基于协议ReCombine的秘密共享物重新分配阶段的算法。

[0017] 图12示意性地示出了SPDZ协议架构对比所提出的容错SPDZ协议架构。

[0018] 图13描绘了示出SPDZ协议和所提出的容错SPDZ协议的比较的表格。

具体实施方式

[0019] 为了促进对本公开原理的理解的目的，现在将参考附图中图示的和以下书面说明书中描述的实施例。理解到，不意图借此对本公开范围的任何限制。进一步理解到，本公开包括对所图示实施例的任何更改和修改，并且包括如本公开所属领域的普通技术人员通常将想到的本公开原理的另外应用。

[0020] 本公开针对一种用于利用SPDZ计算多方内积的高效方法。以在当事方间共享附加的预计算数据为代价，根据本公开的方案能够针对SPDZ协议的在线阶段将乘法以及所需要的Beaver三元组的数量减少一半，从而在利用SPDZ计算多方内积时显著改进在线阶段和离线阶段二者的性能。

[0021] 如本文中所使用的，术语“Beaver三元组”指代三个值的集合(a,b,c)，其中a和b是随机均匀选择的，并且c等于a乘以b的乘积：$c = ab$。Beaver三元组中的所有三个数字是有限域F_p的成员，其中p是模数值。在其中$p = 2^n - 1$的一些说明性实施例中，有限域F_p包括$[-2^{n/2}, 2^{n/2} - 1]$范围中的所有整数，其中n是64、80或128，虽然也可以使用其他的有限域。如下面进一步详细描述的，在修改的SPDZ协议中，当事方在离线计算阶段期间生成Beaver三元组，并且以安全的方式分配Beaver三元组的共享物以供稍后作为在线计算阶段的部分用于内积的多方计算。

[0022] 如本文中所使用的，术语“同态”和“同形”指代一些密码系统的属性，在所述密码系统中，数学运算可以应用于加密值以产生与对明文值执行相同运算的结果相匹配的加密结果，而不将加密值或结果的内容透露给执行运算的计算设备。例如，给定两个值$a = 1$和b

= 2及其对应的加密配对部*a'*和*b'*，在提供加性同态的密码系统中，加密值*c'* = *a'* + *b'*存储数字"3"的加密表示(1 + 2 = 3)。值*c'*可以由不能确定*a'*或*b'*的明文值的计算机来计算，并且在一些实例中，产生*c'*的计算机可以在没有用于确定*a*或*b*的原始明文值的能力的情况下解密*c'*以检索明文总和(3)。减性同态类似地应用于将两个加密值相减的运算(例如，*a* = 2，*b* = 1，*c'* = *a'* - *b'*，解密(*c'*) = 1)，并且在本文中描述的实施例中，减性同态也等效于加性同态，在所述加性同态中，一个或多个所选值被取负以执行等效于减法的加法运算。类似地，对于提供乘性同态的密码系统，两个加密值(*a'*和*b'*)的乘积是解密成与两个明文值(*a*和*b*)的乘积相同的值的另一个加密值(*c'*)(例如*a* = 2，*b* = 3，*c'* = (*a'b'*)，解密(*c'*) = 6)。在本文中描述的实施例中，⊞符号指代在提供加性同态的密码系统中的两个加密值之间的加法，⊟指代在提供减性同态的密码系统中的两个加密值之间的减法运算，其中减法可以被认为是取负值的加法，并且⊠符号类似地指代在提供乘性同态的密码系统中的两个加密值之间的乘法。

[0023] 如本文中所使用的，项λ指代选自有限域F_p $(\lambda \in F_p)$的消息认证码(MAC)密码密钥。向执行内积运算的每个当事方P_i $(1 \le i \le n)$提供总体秘密λ的个体共享物λ_i，其中λ是所有个体秘密的总和$(\lambda = \lambda_1 + \ldots + \lambda_n)$。没有任何当事方具有对完整的MAC密码密钥$\lambda$的访问。MAC密码密钥$\lambda$可以经由乘法生成与针对值$x$的消息认证码相对应的加密MAC值$\lambda x$，并且如下面所描述的，MAC密钥$\lambda$本身可以使用公钥被加密以生成可以在多个当事方间共享的MAC密钥的加密密文ct_λ。在多方计算操作期间，加密的MAC值用于对从其他当事方接收的信息的验证，并且提供针对更改共享数据的恶意当事方的抵抗。本文中描述的一些实施例可以省略MAC密钥λ在多方计算系统的配置中的使用，该多方计算系统在多方计算期间保护原始输入数据的隐私但是信任参与者提供准确的数据输出。

[0024] 如本文中所使用的，项$\varepsilon = (Enc, Dec)$指代至少提供具有阈值解密的加性/减性和乘性同态的非对称公钥加密系统。项*sk*指代ε中用于利用*Dec*函数的密文解密的私钥，并且项*pk*指代ε中用于明文加密以使用*Enc*函数产生密文的公钥。执行多方内积计算的每个当事方P_i $(1 \le i \le n)$具有对私钥sk_i的部分和完整公钥*pk*的访问，其中$sk = sk_1 + \ldots + sk_n$，其中没有任何当事方具有对完整私钥*sk*的访问。

[0025] 如本文中所使用的，针对值的符号[ ](例如[*x*])指示该值的"共享物"或者其中多个当事方各自持有总体值*x*的共享物[*x*]，但是具有对共享物[*x*]的访问并不透露*x*的实际值。在下面描述的实施例中，值*x*是加密的密文值，并且本文中使用的密码系统的加性同形属性使得总体值*x*能够被划分成具有*x*的总体值的个体共享物。在来自具有多个元素的两个向量的内积计算中，向量的每个元素表示以安全方式在两个或更多个节点间共享的值。在数学上，对于总共*n*个当事方P_i $(1 \le i \le n)$的$[x] = \{(x_1, \gamma_1(x)), \ldots, (x_n, \gamma_n(x))\}$，其中每个共享物$[x_i]$是总体[*x*]的部分，其中$x = x_1 + x_2 \ldots + x_n$，针对每个当事方的项$\gamma_i(x)$是值$H_x$的另一个个体共享物，所述值$H_x$是密码密钥$\lambda$乘以值*x*的输出($H_x = \lambda x$)，并且给定个体共享物$\gamma_i(x)$，总计为$H_x = \gamma_1(x) + \gamma_2(x) + \ldots + \gamma_n(x)$。每个当事方将共享值[*x*]存储为元组中的一个$(x_i, \gamma_i(x))$，并且不直接将该元组透露给群组中的其他当事方。

[0026] SPDZ协议使用两阶段方案，其包括多个当事方用于与彼此安全交换被称为Beaver三元组的随机生成的数字的计算的"离线"阶段。Beaver三元组不是对节点感兴趣的实际数

据，而是在后续在线阶段中用于使能实现共享结果的计算同时防止每个当事方标识来自一个或多个附加当事方的私有数据。离线阶段引入计算以计算和共享包括加密操作的Beaver三元组，并且一般被认为是SPDZ协议在计算上更密集的阶段。在第二“在线”阶段中，当事方使用Beaver三元组和感兴趣的实际数据来执行安全多方计算，该安全多方计算包括上面描述的内积计算。虽然SPDZ协议在执行内积计算时是有效的，但由于SPDZ针对多方乘法而实现的基础操作，在在线阶段期间在内积计算中发生的大量乘法运算引入了附加的通信操作——每个通信操作要求Beaver三元组的使用——并且在当事方之间产生计算开销。

[0027] 在离线阶段和在线阶段之前，必须生成密码密钥，包括生成公共密码参数、公钥、私钥共享物(由每个当事方拥有)和MAC密钥共享物(由每个当事方拥有)。特别地，每个当事方接收n个MAC密钥共享物中的一个，其中n是当事方的总数。假设λ是MAC密钥并且$\lambda_1 + \cdots + \lambda_n$是秘密(MAC)共享物，那么当事方$P_i$($1 \le i \le n$)拥有秘密共享物$\lambda_i$。应指出，除非需要更新MAC密钥，否则该步骤仅执行一次。

[0028] 在SPDZ协议的离线阶段期间，生成将允许当事方计算乘积的足够数量的乘法三元组，特别是Beaver乘法三元组。Beaver乘法三元组由三个值([a],[b],[c])组成，其分别是a、b、c的共享物，其中$a, b, c \in F_p$是随机值且$c=ab$。这些值可以由密码提供者生成，并且于在线阶段之前将所得到的共享物分配给当事方。

[0029] 给定针对某值s的密文$\mathrm{ct}_s = \mathrm{Enc}_{\mathrm{pk}}(s)$，SPDZ协议Share可以安全地分配$s$的秘密共享物，如图1中所示。协议$\mathrm{Share}(\mathrm{ct}_s, P_1, \ldots, P_n)$，其中$\mathrm{ct}_s = \mathrm{Enc}_{\mathrm{pk}}(s)$是公共已知的并且$P_1, \ldots, P_n$是$n$个当事方。参考图1，每个当事方随机均匀生成一值，值$f_i \in F_p$，并且将其加密成密文$\mathrm{ct}_{f_1} = \mathrm{Enc}_{\mathrm{pk}}(f_i)$(块100)。然后所有当事方广播其密文使得每个当事方可以计算密文$\mathrm{ct}_{s+f} = \mathrm{ct}_s \boxplus \mathrm{ct}_{f_1} \boxplus \ldots \boxplus \mathrm{ct}_{f_n}$(块102)。然后所有当事方利用其自己的私钥$sk$的秘密共享物协作地解密$\mathrm{ct}_{s+f}$并且获得$s+f$，其中$f = \sum_{i=1}^{n} f_i$(块104)。当事方$P_1$将$s$的秘密共享物设置为$s + f - f_1$并且当事方$P_i (i \neq 1)$将$s$的秘密共享物设置为$-f_i$(块106)。

[0030] 利用协议Share，令$\mathrm{ct}_\lambda = \mathrm{Enc}_{\mathrm{pk}}(\lambda)$为MAC密钥$\lambda$的密文，然后执行生成三元组([$a$],[$b$],[$c$])的过程。可以使用以下步骤生成[$a$]：

- 当事方P_i随机均匀选择值$a_i \in F_p$，并且将其加密成$ct_{a_i} = \mathrm{Enc}_{\mathrm{pk}}(a_i)$
- 一个所选当事方计算$\mathrm{ct}_a = \mathrm{ct}_{a_1} \boxplus \ldots \boxplus \mathrm{ct}_{a_n}$以及$\mathrm{ct}_{\lambda a} = \mathrm{ct}_\lambda \boxtimes \mathrm{ct}_a$。
- 所有当事方一起执行$\mathrm{Share}(\mathrm{ct}_{\lambda a}, P_1, \ldots, P_n)$，使得当事方$P_i$($1 \le i \le n$)持有$\lambda a$的秘密共享物$\gamma_i(a)$。因此，当事方$i$持有$(a_i, \gamma_i(a))$，其中$a = a_1 + \cdots + a_n$并且$\lambda \times a = \gamma_1(a) + \cdots + \gamma_n(a)$。

[0031] 重复用于生成[a]的相同步骤以生成[b]。生成值[c]使得$c = ab$。给定来自步骤1和步骤2的ct_a和ct_b，令$\mathrm{ct}_c = \mathrm{ct}_a \boxtimes \mathrm{ct}_b$。执行协议$\mathrm{Share}(\mathrm{ct}_c, P_1, \ldots, P_n)$，使得当事方$i$持有$c$的秘密共享物$c_i$。给定$c + f$，当对$\mathrm{ct}_c$执行协议Share时，可以计算新的密文$\mathrm{ct'}_c = \mathrm{Enc}_{\mathrm{pk}}(c + f) \boxminus \mathrm{ct}_{f_1} \boxminus \ldots \boxminus \mathrm{ct}_{f_n}$以及$\mathrm{ct}_{\lambda c} = \mathrm{ct}_\lambda \boxtimes \mathrm{ct}_c$，其中$\mathrm{Enc}_{\mathrm{pk}}(c + f)$是通过加密$c + f$的新密文。所有当事方一起执行$\mathrm{Share}(\mathrm{ct}_{\lambda c}, P_1, \ldots, P_n)$，使得当事方$P_i$($1 \le i \le n$)持有

λc的秘密共享物$\gamma_i(c)$。最终，当事方i持有$(c_i, \gamma_i(c))$，使得$c = c_1 + \cdots + c_n$并且$\lambda c = \gamma_i(c) + \cdots + \gamma_n(c)$。

[0032] 如本领域中已知的，生成三元组要求执行公钥操作(即，在密文之上的同形加法和乘法)，并且因此对于大量三元组而言离线阶段是耗时的。应指出，beaver乘法三元组是一种元组；诸如反元组([x]，[1/x])的若干其他元组也出于不同目的而存在于协议SPDZ中。

[0033] 由于任何运算电路都可以表示为加法和乘法门的集合，因此在线阶段聚焦于如何安全地计算加法和乘法。多方加法使得多个当事方能够将l个不同值的不同共享物相加以生成总和，并且多方加法还包括通过将适当值取负来相减以执行减法。在图2中描绘了SPDZ协议Add的实施例。在其中$l = 2$的简单示例中，多方加法将两个数字的共享物相加在一起，虽然多方加法也可以将多于两个数字的共享物相加。项$[x^1], [x^2], \ldots, [x^l]$分别表示$l$个不同值$x^1, x^2, \ldots, x^l$的共享物，所述共享物在多方加法过程中相加。在本文档的上下文中，上标符号不指示数学上的指数。如上面描述的，每个共享物包括元组$(x_i, \gamma_i(x))$。在多方加法期间，每个当事方基于来自所有共享物的x_i项的总和来计算值z_i：$z_i = x_i^1 + x_i^2 + \ldots + x_i^l$，并且基于来自所有共享物的$\gamma_i(x)$项的总和来计算值$\gamma_i(z)$：$\gamma_i(z) = \gamma_i(x^1) + \gamma_i(x^2) + \cdots + \gamma_i(x^l)$。这些运算使得每个当事方$P_i (1 \le i \le n)$能够具有总体总和值$z$的共享物$[z_i]$和MAC值$\mathbf{ct}_\lambda z$的共享物$\gamma_i(z)$，其中每个当事方持有元组$(z_i, \gamma_i(z))$。如上面指出的，每个当事方具有总和$z$的共享物$[z_i]$，但是每个当事方不具有对$z$的实际总和值的访问，这确保即使在双方加法过程中，在给定总和以及相加在一起的两个值中的一个值的明文的情况下，一个当事方也无法确定由另一当事方持有的值。

[0034] 在上面描述的多方加法中，针对每个加法的计算复杂度对于相加在一起的l个共享物是$\mathcal{O}(l)$，并且计算包括相对简单的加法运算而对于进一步的加密操作不要求公钥pk的使用。此外，在已经将共享物$[x^1], [x^2], \ldots, [x^l]$分配给当事方之后，不进一步需要当事方之间的通信或Beaver三元组的使用来针对每个当事方计算总和的共享物$[z_i]$。

[0035] 多方乘法使能实现具有两个值x和y的共享物$[x]$和$[y]$的当事方。图3中的协议Mult示出了如何安全地执行乘法。与可以将来自多于两个值的共享物相加在一起的多方加法不同，多方乘法应用于恰好两个值。多方乘法过程使用在下面描述的离线阶段中已经共享的Beaver三元组$([a], [b], [c])$。在多方乘法过程期间，每个当事方$P_i (1 \le i \le n)$使用上面描述的加法过程来将输入的共享物$[x]$和取负的Beaver三元组共享值$[a]$相加：$Add([x_i], -[a_i])$。类似地，每个当事方使用上面描述的加法过程来将输入的共享物$[y]$和取负的Beaver三元组共享值$[b]$相加：$Add([y_i], -[b_i])$。每个当事方向所有其他当事方广播上面描述的两个加法的结果：$\epsilon = x - a$和$\rho = y - b$。如上面所描述的，由于加性同态，结果ε和ρ透露x和a的元组与y和b的元组之间的总差异。在每个当事方生成两个本地值z_i和$\gamma_i(z)$时，多方乘法过程继续使用来自Beaver三元组中每个共享物$[c]$的c_i值：

$$z_i = \begin{cases} c_i + \epsilon b_i + \rho a_i + \epsilon\rho & (i = 1) \\ c_i + \epsilon b_i + \rho a_i & (i \neq 1) \end{cases}$$和

$$\gamma_i(z) = \gamma_i(c) + \epsilon\gamma_i(b) + \rho\gamma_i(a) + \lambda_i\epsilon\rho。$$

[0036] 多方乘法过程使得每个当事方能够持有原始值x和y的乘积$(z = xy)$的共享物$[z_i]$

，其中再一次$[z_i] = (z_i, \gamma_i(z))$。与多方加法过程一样，即使仅存在参与多方乘法的两个当事方并且假定每个当事方具有对明文输入x或y中的一个的访问，来自乘法过程的共享物$[z_i]$也防止任何一个当事方标识由另一当事方持有的值，这是因为共享物$[z_i]$不透露真实的乘积值z。多方乘法过程要求当事方之间的二者通信以产生共享值ε和ρ。此外，乘法过程消耗一个Beaver三元组的共享物($[a]$,$[b]$,$[c]$)，其中每个Beaver三元组仅被使用一次。

[0037] 图4描绘了包括多个计算节点104A和104B的系统100，所述多个计算节点104A和104B经由数据网络150彼此通信连接。在系统100中，节点104A和104B中的每个是在本文中描述的MPC过程中的一个当事方。每个节点是在本文中描述的安全多方计算过程中充当单个当事方的计算设备，并且术语“节点”在下面描述的实施例中与术语“当事方”可互换使用。图4详细描绘了节点104A，并且以类似的方式利用不同的存储数据集合配置节点104B，以使用本文中描述的实施例实现安全多方内积计算。虽然图4出于说明性目的描绘了具有两个节点104A-104B的系统，但是本文中描述的实施例也可以由三个或更多个节点的群组来执行。

[0038] 更详细地参考节点104A，该节点包括操作性地连接到网络接口设备112和存储器120的处理器108。处理器108典型地是具有一个或多个处理核的中央处理单元(CPU)，所述一个或多个处理核执行在存储器120中存储的程序指令124以实现本文中描述的实施例。然而，处理器108的其他实施例代替或附加于以下各项而使用不同的处理元件：包括图形处理单元(GPU)的CPU、现场可编程门阵列(FPGA)、专用集成电路(ASIC)、数字信号处理器(DSP)以及被配置为执行本文中描述的操作的任何其他数字逻辑器件。在一些实施例中，处理器108实现硬件随机数生成器(RNG)或使用软件随机数生成器或伪随机数生成器(PRNG)来生成随机值。在本文中的描述中，对随机值的生成或随机值的使用的任何提及指代RNG或PRNG以均匀随机的方式生成选自预定数值范围(例如有限域F_p)的值的操作。

[0039] 网络接口设备112将节点104A连接到诸如局域网(LAN)或广域网(WAN)之类的数据网络150以使能图4的节点104A与节点104B之间的通信，以及连接到执行本文中描述的安全多方内积计算过程的附加节点。网络接口设备112的非限制性实施例包括诸如以太网适配器之类的有线网络设备或者诸如无线LAN或无线WAN网络适配器之类的无线网络设备。在系统100中，虽然节点104A和104B可以使用防止窃听者观察或更改经由网络150的通信的传输层安全(TLS)加密和认证的信道或其他等效通信信道进行通信，但是假定通过网络150进行的所有数据传输由可以记录从节点104A和104B传输的所有数据的第三方对抗者(图4中未示出)观察到。附加于在多方内积计算期间防止节点104B标识节点104A的私有明文数据128并且防止节点104A标识节点104B的私有明文数据128，本文中描述的实施例还防止可以观察到节点104A与104B之间的通信的对抗者确定任何私有数据的值。

[0040] 存储器120包括诸如随机存取存储器(RAM)之类的一个或多个易失性存储器设备和诸如磁盘或固态存储器设备之类的非易失性存储器设备，所述一个或多个易失性存储器设备和非易失性存储器设备存储程序指令124、私有明文数据128、值共享物数据132、密码密钥数据163和内积的计算结果140。私有明文数据128包括形成对内积计算的输入的两个向量中的一个向量的全部或部分。明文数据128中的两个向量在本文中称为x和w。值共享物数据132包括每个节点在系统100的操作期间接收的共享物，其包括在离线阶段期间生成的Beaver三元组值的共享物和在内积计算过程期间交换的值的共享物。如上面描述的，每个

共享物向每个节点提供每个值的部分的加密版本。密码密钥数据136包括用于节点104A的加密MAC密码密钥的共享物、公钥pk以及与节点104A共享的私钥sk_i的共享物。内积的计算结果140是内积值的多方计算的最终输出。

[0041] 节点104B包括具有类似数据结构的存储器，除了节点104B存储与用于生成内积的第二向量相对应的私有明文数据128的不同集合、秘密值共享物132的不同集合和密码密钥数据136中的私钥和MAC密钥的不同共享物。如下面所描述的，节点104A和104B二者计算内积的相同结果140。在图4的说明性实施例中，节点104A存储向量x的所有明文元素，并且节点104B存储向量w的所有明文元素，但是在其他实施例中，用于这些向量中的任一个或二者的明文数据被分配给多个节点，以使得大量节点能够执行安全多方内积计算。

[0042] 具体来说，在我们的设置中，一个当事方持有输入向量$x = \{x_1, \ldots, x_N\}$并且另一当事方持有权重向量$w = \{w_1, \ldots, w_N\}$，并且它们希望联合计算以下等式(等式1)：

$$x_1 w_1 + \cdots + x_N w_N = \sum_{i=1}^{N} x_i w_i \quad (1)。$$

[0043] 以下是用于安全内积计算的先前已知的协议的描述。这里我们仅聚焦于在线阶段，并且我们假定已经关于SPDZ协议设立了离线阶段。

[0044] 先前已知的内积是由协议(IP)$([P_1, x_1, \ldots, x_N], [P_2, w_1, \ldots, w_N], \mathrm{ct}_\lambda)$给出的，其中当事方$P_1$持有私有向量$x_1, \ldots, x_N$，并且当事方$P_2$持有私有向量$w_1, \ldots, w_N$并且$\mathrm{ct}_\lambda$是MAC密钥$\lambda$的密文并且仅能由两个当事方一起解密。

1:对于$i = 1, \ldots, N$，当事方1随机均匀选择值x_{i2}，设置$x_{i1} = x_i - x_{i2}$，生成$\mathrm{ct}_{\lambda x_i} = \mathrm{Enc}_{\mathrm{pk}}(x_i) \boxtimes \mathrm{ct}_\lambda$，将$x_{i2}$发送到当事方2并且运行协议$\mathrm{ReShare}(\mathrm{ct}_{\lambda x}, P_1, P_2)$，使得$x_i$在当事方$P_1$与当事方$P_2$之间安全共享，其由$[x_i]$标示。

2:对于$i = 1, \ldots, N$，当事方2随机均匀选择值w_{i1}，设置$w_{i1} = w_i - w_{i2}$，生成$\mathrm{ct}_{\lambda w_i} = \mathrm{Enc}_{\mathrm{pk}}(w_i) \boxtimes \mathrm{ct}_\lambda$，将$w_{i1}$发送到当事方1并且运行协议$\mathrm{ReShare}(\mathrm{ct}_{\lambda x}, P_1, P_2)$，使得$x_i$在当事方$P_1$与当事方$P_2$之间安全共享，其由$[w_i]$标示。

3:当事方P_1和当事方P_2运行SPDZ协议$(\mathrm{Mult}[x_i], [w_i])$$(1 \le i \le N)$，其被标示为$[d_i]$。

4:当事方P_1和当事方P_2然后执行SPDZ协议$(\mathrm{Add}[d_i], \ldots, [d_N])$。

[0045] 如上面指出的，在协议期间发生的一个操作是安全重新共享操作。被称为“ReShare”或“ReSharing”操作或协议的安全重新共享操作使得两个或更多个节点能够生成对所有节点都已经知道的值的共享物(即，完整的值已经被节点“共享”，并且该过程“重新共享”总体值的部分)，其中每个节点持有总体值的共享物，而不知道由任何其他节点持有的共享物。在本文中描述的实施例中，重新共享操作被定义为：$Reshare\ (ct_s, P_1, \ldots, P_n)$其中$ct_s$是秘密值$s$的加密密文$(ct)$，并且$P_1 \ldots P_n$表示参与重新共享操作的$n$个当事方。加密的密文值$ct$作为整体对所有当事方已知，并且重新共享操作使得当事方能够各自接收原始数据集合的共享物而不知道其他节点的共享物，甚至在双节点系统中也是如此。重新共享操作开始于使用公钥pk对秘密值s加密以产生密文$ct_s = Enc_{pk}(s)$，并且密文ct_s被传输到接收秘密值s的加密共享物的当事方群组中的所有n个当事方。n个当事方中的每个当事方i生成均匀随机值$f_i \in F_p$，并且使用公钥pk对随机值加密以生成随机值的密文：$ct_{f_i} = Enc_{pk}(f_i)$。每个当事方将相应的密文值ct_{f_i}经由数据网络传输到所有其他当事方，并且每个当事方计算s的加密值和个体加密值ct_{f_i}的密文总和ct_{s+f}：$ct_{s+f} = c_s \boxplus \mathrm{ct}_{f_1} \boxplus \mathrm{ct}_{f_2} \boxplus \ldots \boxplus \mathrm{ct}_{f_n}$

，其中，⊞运算符再一次指示在提供加性同态的密码系统中的密文的加法。然后，当事方使用私钥ski的个体部分来执行协作解密过程，以对值ct_{f_i}进行解密来获得$s + f$的总和，其中f是所有个体随机值f_i的总和。n个当事方间的最终共享物对于当事方1是$[s_1] = s + f - f_1$并且对于其他i个当事方中的每个是$[s_i] = -f_i$。

[0046] 上面描述的重新共享操作是一种用以分配密文值的共享物的方法，该密文值对群组中的所有当事方都是已知的。另一种操作被称为“共享物分配”操作，该操作生成一明文值的共享物，该明文值对群组中的一个当事方是已知的，并且该明文值在群组中的节点之间被共享之前不需要被加密而同时不向其他当事方公开明文数据。如下所述，该操作使得执行多方内积计算的系统中的节点能够基于单个向量中的两个元素的乘法来分配对乘积和向量中的个体元素的共享物，其中一个节点具有对元素的明文内容的访问。给定被称为x的明文值，拥有值x的当事方针对群组中每个其他当事方来使用F_p中的均匀随机分布来生成随机值，从原始值x中减去所有随机生成的值以生成由原始当事方使用的共享物$[x]$，并且将随机生成的值传输给一个或多个附加当事方，其中随机值用作由其他当事方使用的$[x]$的共享物。例如，在具有两个当事方的群组中，第一当事方在存储器中持有x_i，其中i指示向量中值的索引，并且生成被标注为x_{i2}的随机值，其中2是指示接收随机值的另一个节点的索引标注。第一当事方基于x_i与x_{i2}之间的差生成被标注为$[x_{i1}] = x_i - x_{i2}$的x_i的共享物。第一当事方将值x_{i2}传输到第二当事方，并且该值是由第二当事方接收和持有的x_i的共享物($[x_{i2}]$)。由于值x_{i2}仅仅是与原始值x_i无关的随机值，因此该随机值可以通过数据网络传输并且被对抗者和群组中的所有当事方观察到，而不损害x_i的隐私。通过生成附加随机值(每个附加当事方一个)，并且由第一当事方从原始值中减去所有随机值以产生每个当事方的共享物，可以将共享物分配操作扩展到三个或更多个当事方。如上所述，因为每个其他当事方仅接收随机值，所以值x不需要被加密来执行共享分配操作，尽管在x被加密的情况下共享分配操作也可以被应用于加密值。重新共享和共享分配操作二者均产生由多个当事方持有的值的共享物。重新共享操作附加地防止单个当事方具有关于其他当事方持有的共享物的信息，而共享物分配操作使得持有原始明文值的第一当事方能够具有关于所有共享物的信息，但是使得其他当事方不能够观察到第一当事方的共享物。

[0047] 使用SPDZ的先前已知的内积运算将需要执行协议ReShare $2N$次、协议Mult N次以及协议Add一次。此外，先前已知的内积运算消耗在离线阶段中生成的N个三元组，因为三元组的数量与所需要的乘法(Mult)数量线性相关。

[0048] 本公开提出了一种对安全内积计算的优化方案，该方案可以减少所需要的乘法数量，并且因此减小所使用的三元组的数量。优化方法的基本思想基于以下观察：

$$(w_1 + x_2)(w_2 + x_1) - x_1x_2 - w_1w_2$$

$$= w_1w_2 + w_1x_1 + w_2x_2 + x_1x_2 - x_1x_2 - w_1w_2。$$

$$= x_1w_1 + x_2w_2$$

[0049] 如果可以很好地建立$[w_1], [x_1], [x_2], [x_1x_2]$和$[w_1w_2]$，那么仅需要一个乘法(即，$w_1 + x_2$与$w_2 + x_1$的乘法。幸运地，由于当事方$P_1$持有$x_1$和$x_2$，那么可以以类似于$x_1$或$x_2$的方式分配$[x_1x_2]$。此外，由于当事方$P_2$持有$w_1$和$w_2$，那么可以以类似于$w_1$或$w_2$的方式分配$[w_1w_2]$。

[0050] 基于上面的观察，可以提供优化的安全内积计算协议(协议IP^+)。该协议在内积计算的在线阶段中引发$N/2$个乘法运算，并且因此也将三元组的数量减少到$N/2$。

[0051] 协议$IP^+([P_1,x_1,\ldots,x_N],[P_2,w_1,\ldots,w_N])$，其中当事方$P_1$持有私有向量$x=\{x_1,\ldots,x_N\}$，并且当事方$P_2$持有私有向量$w=\{w_1,\ldots,w_N\}$。为简单起见，假定$N=2k$。

1:对于$i=1,\ldots,N$，当事方P_1随机均匀选择值x_{i2}，设置$x_{i1}=x_i-x_{i2}$，生成$\mathrm{ct}_{\lambda x_i}=\mathrm{Enc}_{\mathrm{pk}}(x_i)\boxtimes\mathrm{ct}_\lambda$，将$x_{i2}$发送到当事方2并且运行协议$\mathrm{ReShare}(\mathrm{ct}_{\lambda x},P_1,P_2)$，使得$x_i$在当事方$P_1$与当事方$P_2$之间安全共享，其由$[x_i]$标示。

2.对于$i=1,\ldots,k$，当事方P_1计算$\overline{x_i}=x_{2i-1}x_{2i}$，随机均匀选择值$\overline{x_{i2}}$，设置$\overline{x_{i1}}=\overline{x_i}-\overline{x_{i2}}$，生成$\mathrm{ct}_{\lambda\overline{x_i}}=\mathrm{Enc}_{\mathrm{pk}}(\overline{x_i})\boxtimes\mathrm{ct}_\lambda$，将$\overline{x_{i2}}$发送到当事方2并且运行协议$\mathrm{ReShare}(\mathrm{ct}_{\lambda\bar{x}},P_1,P_2)$，使得$\overline{x_i}$在当事方$P_1$与当事方$P_2$之间安全共享，其由$[\overline{x_i}]$标示。

3:对于$i=1,\ldots,N$，当事方2随机均匀选择值w_{i1}，设置$w_{i1}=w_i-w_{i2}$，生成$\mathrm{ct}_{\lambda w_i}=\mathrm{Enc}_{\mathrm{pk}}(w_i)\boxtimes\mathrm{ct}_\lambda$，将$w_{i1}$发送到当事方1并且运行协议$\mathrm{ReShare}(\mathrm{ct}_{\lambda x},P_1,P_2)$，使得$x_i$在当事方$P_1$与当事方$P_2$之间安全共享，其由$[w_i]$标示。

4.对于$i=1,\ldots,k$，当事方P_2计算$\overline{w_i}=w_{2i-1}w_{2i}$，随机均匀选择值$\overline{w_{i1}}$，设置$\overline{w_{i1}}=\overline{w_i}-\overline{w_{i2}}$，生成$\mathrm{ct}_{\lambda\overline{x_i}}=\mathrm{Enc}_{\mathrm{pk}}(\overline{w_i})\boxtimes\mathrm{ct}_\lambda$，将$\overline{w_{i1}}$发送到当事方1并且运行协议$\mathrm{ReShare}(\mathrm{ct}_{\lambda\bar{w}},P_1,P_2)$，使得$\overline{w_i}$在当事方$P_1$与当事方$P_2$之间安全共享，其由$[\overline{w_i}]$标示。

5.对于$i=1,\ldots,k$，当事方P_1和当事方P_2运行SPDZ协议$\mathrm{Add}([w_{2i-1}],[x_{2i}])$和$\mathrm{Add}([w_{2i}],[x_{2i-1}])$，其分别得到$(w_{2i-1}+x_{2i})$和$(w_{2i}+x_{2i-1})$的共享物$[w_{2i-1}+x_{2i}]$和$[w_{2i}+x_{2i-1}]$。当事方$P_1$和当事方$P_2$然后执行协议$\mathrm{Mult}([w_{2i-1}+x_{2i}],[w_{2i}+x_{2i-1}])$，其结果被标示为$[d_i]$。

6.当事方P_1和当事方P_2运行协议$\mathrm{Add}([d_1],\ldots,[d_k],-[\overline{x_1}],\ldots,-[\overline{x_k}],-[\overline{w_1}],\ldots,-[\overline{w_k}])$以确定内积$(d)$。

[0052] 我们观察到，IP^+以共享N个更多私有值的成本将协议Mult的必要执行数量从N减少到$N/2$。然而，执行协议Add的成本可以忽略，因为它是在本地运行的并且与协议Mult相比可以忽略不计)。然而，应指出，共享一个私有值仅需要执行一次协议Reshare，但是生成一个三元组需要执行4次协议Reshare。此外，值得指出的是，执行Mult需要当事方之间的交互(以便透露协议Mult中示出的中间值ε和ρ)，将Mult的执行数量从N减少到$N/2$也减少了通信和计算复杂度。由于协议Add的计算复杂度与输入是线性的，因此IP需要$2N-2$个算术加法，而IP^+引发$5N-2$个算术加法。

[0053] 本公开的其余部分将针对与SPDZ协议以及同上述方法一起使用或可以同上述方法一起使用的其他相关协议有关的背景信息。

[0054] 协议SPDZ中的秘密共享方案是n中取n加性秘密共享方案。根据加性秘密共享方案，给定秘密值α和当事方集合$\{P_1,\ldots,P_n\}$，随机均匀选择$\alpha_1,\ldots,\alpha_{n-1}$，并且选择$\alpha_n$使得它满足$\alpha_1+\cdots+\alpha_n=\alpha$，其中$\alpha_i$被分配给当事方$P_i$。因此，需要$n$个当事方(即所有当事方)以便恢复秘密值$\alpha$。

[0055] 为了容忍至多$n-t$个不可用的当事方，t个可用的当事方应当能够恢复由那些$n-t$个不可用当事方持有的秘密共享物。基于该观察，提出了:协议ReShare，其中每个当事

方将其持有的秘密共享物重新分配给t个当事方的任何子集;以及协议ReCombine,其中t个当事方可以恢复不可用当事方的秘密共享物,并且然后如果需要,则最终重建秘密值。基于这两个协议,提出了秘密共享物重新分配阶段和秘密共享物重新组合阶段,这两个阶段可以放在协议SPDZ的离线阶段与在线阶段之间,使得所得到的SPDZ协议享有容错属性。

[0056] 在图5中描绘了协议ReShare。第一循环经由认证的机密信道将每个当事方的持有秘密共享物重新分配给所有可用当事方。第二循环对重新分配的秘密共享物求和,并且针对每个当事方将总和与被指派给每个当事方的唯一标识符相关联地存储。这里的聚合操作减少存储开销。应指出,标注字符串"Label"用于唯一地标识子集中的当事方以及相关联的重新分配的秘密共享物,并且可以促进重建秘密共享物。值得指出的是,任何秘密共享物以加性方式被重新分配给t个当事方的子集,因此任何$t-1$个当事方(除了持有该秘密共享物的当事方之外)不能恢复该秘密共享物。给定n个当事方,具有t个当事方的子集的数量是$\binom{n}{t}$;因此,包括一个特定当事方的子集的数量是$\binom{n}{t}/n=\binom{n-1}{t}$。因此,每个当事方需要存储$\binom{n-1}{t}$个重新分配的秘密共享物和相应的标注。

[0057] 在图6中描绘了协议ReCombine。协议ReCombine使得t个当事方能够使用其持有的秘密共享物和重新分配的秘密共享物来计算秘密值的秘密共享物。每个当事方$P_i(1\leq i\leq n)$持有秘密共享物α_i和重新分配的秘密共享物的集合AggSet_i。

[0058] 现在将讨论示例来说明协议ReShare和ReCombine。假设存在分别持有秘密共享物x_1、x_2、x_3和x_4的四个当事方(即,n=4),使得$x=x_1+x_2+x_3+x_4$,其中x是秘密值。假设系统要求容忍至多两个不可用当事方(即,$n-t$=2),这意味着任何t=2个当事方应当能够重建秘密值。

[0059] 图7描绘了呈现在当事方P_1将其持有的秘密共享物x_1重新分配给其他当事方之后的中间结果的表。应指出,"Label"是按顺序连结所有当事方的索引的字符串,并且它可以唯一标识t个当事方的每个子集。在示例中,由于t=2,因此具有2个当事方的所有子集的集合是$\{\{P_1,P_3\},\{P_2,P_4\},\{P_3,P_4\}\}$。同一行中的重新分配的共享物的总和等于秘密共享物$x_1$。例如,$x_{1,2,\mathrm{Label}_{2,3}}+x_{1,3,\mathrm{Label}_{2,3}}=x_{1,2,\mathrm{Label}_{2,3}}+x_{1,4,\mathrm{Label}_{2,4}}=x_1$并且$x_{1,3,\mathrm{Label}_{3,4}}+x_{1,4,\mathrm{Label}_{3,4}}=x_1$。图8示出了在所有当事方将其持有的秘密共享物重新分配给其他当事方之后的最终结果。

[0060] 图9描绘了在执行协议ReShare之后的最终结果。根据图9,每个当事方持有三个重新分配的秘密共享物和相应的标注以及持有的秘密共享物。因此,在该情况下,任何两个当事方可以构造秘密值。假设当事方P_1和当事方P_2在线但是当事方P_3和当事方P_4不可用。利用协议ReCombine,当事方P_1和当事方P_2分别计算$x_1+x_{3,1,\mathrm{Label}_{1,2}}+x_{4,1,\mathrm{Label}_{1,2}}$和$x_2+x_{3,2,\mathrm{Label}_{1,2}}+x_{4,2,\mathrm{Label}_{1,2}}$。我们可以看到这两个本地计算的值可以恢复秘密值$x$,这是因为$x_{3,1,\mathrm{Label}_{1,2}}+x_{3,2,\mathrm{Label}_{1,2}}=x_3$,$x_{4,1,\mathrm{Label}_{1,2}}+x_{4,2,\mathrm{Label}_{1,2}}=x_4$并且$x_1+x_2+x_3+x_4=x$。

[0061] 本公开提出了基于协议ReShare的秘密共享物重新分配阶段和基于协议ReCombine的秘密重新组合阶段。前一阶段的目的是要让所有当事方将它们持有的秘密共享物重新分配给其他当事方,并且后一阶段的目的是要允许子部分当事方中的当事方重新组合秘密共享物,该秘密共享物可以于在线阶段中使用。

[0062] 在执行离线阶段之后,需要重新分配两个不同种类的数据共享物:密码密钥共享物(即,私钥共享物和MAC密钥共享物)和元组共享物。对于密码密钥共享物,仅考虑MAC密钥

共享物，因为将不于在线阶段中使用私钥共享物。应指出，对于每个元组共享物，由各当事方持有的元组内的每个元素采用形式[x]，所述[x]由数据秘密共享物和MAC值共享物组成。因此，我们仅需要分别在数据秘密共享物和MAC值共享物上应用协议ReShare。图10描述了秘密共享物重新分配阶段，其中重新分配在离线阶段中生成的MAC密钥共享物和元组共享物。

[0063] 在图11中示出的秘密共享物重新组合阶段期间，应用协议ReCombine来计算MAC密钥共享物和元组共享物。应指出，存在多于t个当事方可用是可能的，但仅需要选择t个当事方。

[0064] 如图12中描绘的，给定上面两个阶段，可以通过在离线阶段与在线阶段之间并入上面讨论的秘密共享物重新分配阶段和秘密共享物重新组合阶段来构建容错SPDZ。应指出，在执行秘密共享物重新组合阶段之前，需要选择t个当事方。因此，需要t个当事方是可用的。

[0065] 给定参数n(即，当事方的总数)和t(即，可用当事方的最少数量)，考虑以下度量来将SPDZ协议与容错SPDZ协议进行比较：

- 存储消耗：每个当事方在存储秘密值上所花费的存储消耗。
- 恶意当事方数量：为了确保协议的安全性，允许存在的恶意当事方数量。
- 容错：为了确保协议的正确性，允许存在的可用当事方数量。

[0066] 图13示出了两个协议之间的比较。如可以看到的，针对容错SPDZ的容错特征(即，针对SPDZ的n-t vs. 0)是以增加存储消耗(即，针对SPDZ的$\binom{n-1}{t}+1$vs. n-1)并且减小协议中允许的恶意当事方数量(即，针对SPDZ的t-1 vs. n-1)为代价来被实现的。

[0067] 在上面的章节中，假定参与方不仅执行离线阶段和在线阶段，而且还将其私有输入提供给计算。上面描述的容错SPDZ方案也可以实现为服务，在所述服务中，多个服务提供者托管负责执行离线阶段和在线阶段的许多专用服务器，并且最终用户以秘密共享物的形式上传其私有输入并且将它们存储在多个专用服务器中。应指出，专用服务器中的存储的输入秘密共享物也是采用$[x]=\{(x_1,\gamma_1(x)),\ldots,(x_n,\gamma_n(x))\}$的形式，其中当事方$P_i(1\le i\le n)$持有元组$(x_1,\gamma_1(x))$，使得$x=x_1+\cdots+x_n$并且$\lambda x=\gamma_1(x)+\cdots+\gamma_n(x)$，并且$\lambda$是MAC密钥。因此，在秘密共享物重新分配和秘密共享物重新组合阶段中在私有输入共享物上应用协议ReShare和ReCombine仍然是有效的，使得扩展的SPDZ协议在该模型下享有容错。

[0068] 虽然已经在附图和前述的描述中详细说明和描述了本公开，但是应当将本公开视为在性质方面是说明性的而非限制性的。应理解到，仅呈现了优选实施例，并且期望保护落入本公开的精神内的所有改变、修改和另外的应用。

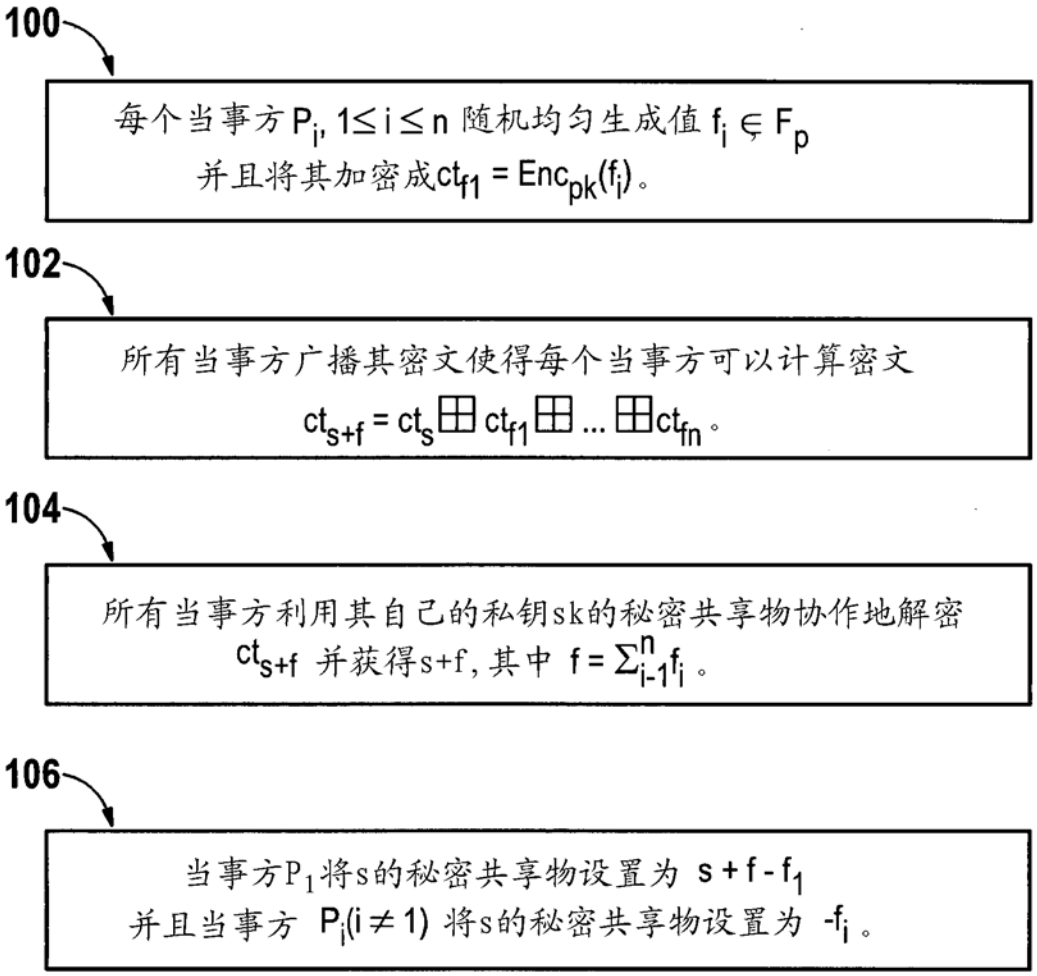

100
每个当事方 P_i, $1 \le i \le n$ 随机均匀生成值 $f_i \in F_p$
并且将其加密成 $ct_{f1} = Enc_{pk}(f_i)$。
102
所有当事方广播其密文使得每个当事方可以计算密文
$ct_{s+f} = ct_s \boxplus ct_{f1} \boxplus \ldots \boxplus ct_{fn}$。
104
所有当事方利用其自己的私钥sk的秘密共享物协作地解密
ct_{s+f} 并获得s+f,其中 $f = \sum_{i-1}^{n} f_i$。
106
当事方 P_1 将s的秘密共享物设置为 $s + f - f_1$
并且当事方 $P_i (i \neq 1)$ 将s的秘密共享物设置为 $-f_i$。

图 1

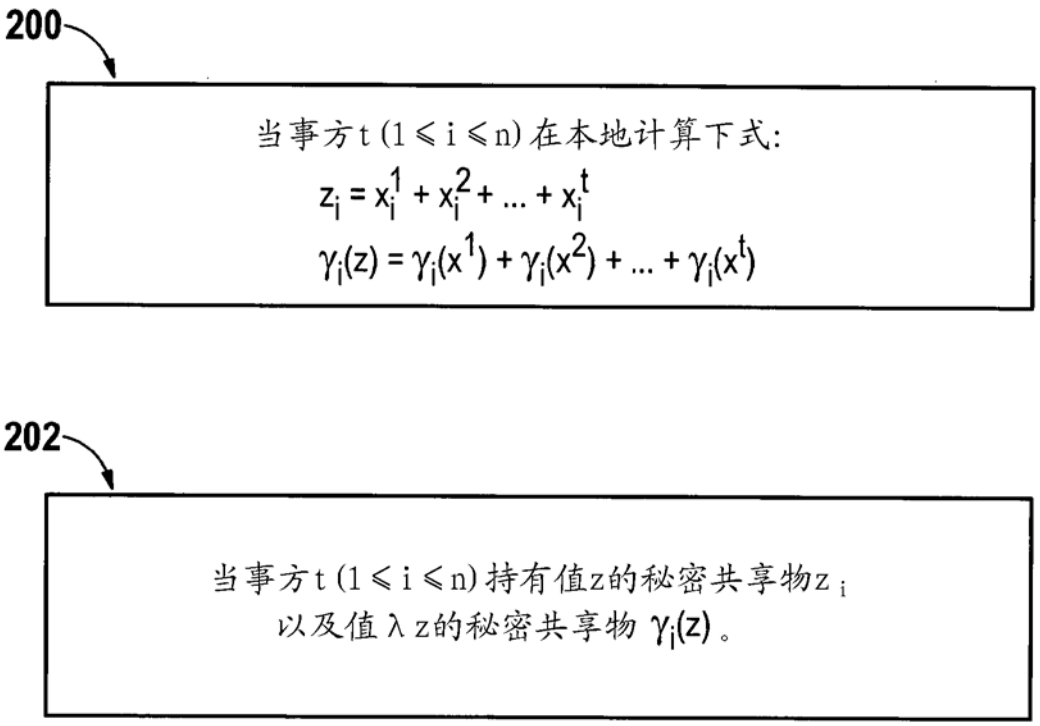

200
当事方t(1≤i≤n)在本地计算下式:
$z_i = x_i^1 + x_i^2 + \ldots + x_i^t$
$\gamma_i(z) = \gamma_i(x^1) + \gamma_i(x^2) + \ldots + \gamma_i(x^t)$
202
当事方t(1≤i≤n)持有值z的秘密共享物z_i
以及值λz的秘密共享物 $\gamma_i(z)$。

图 2

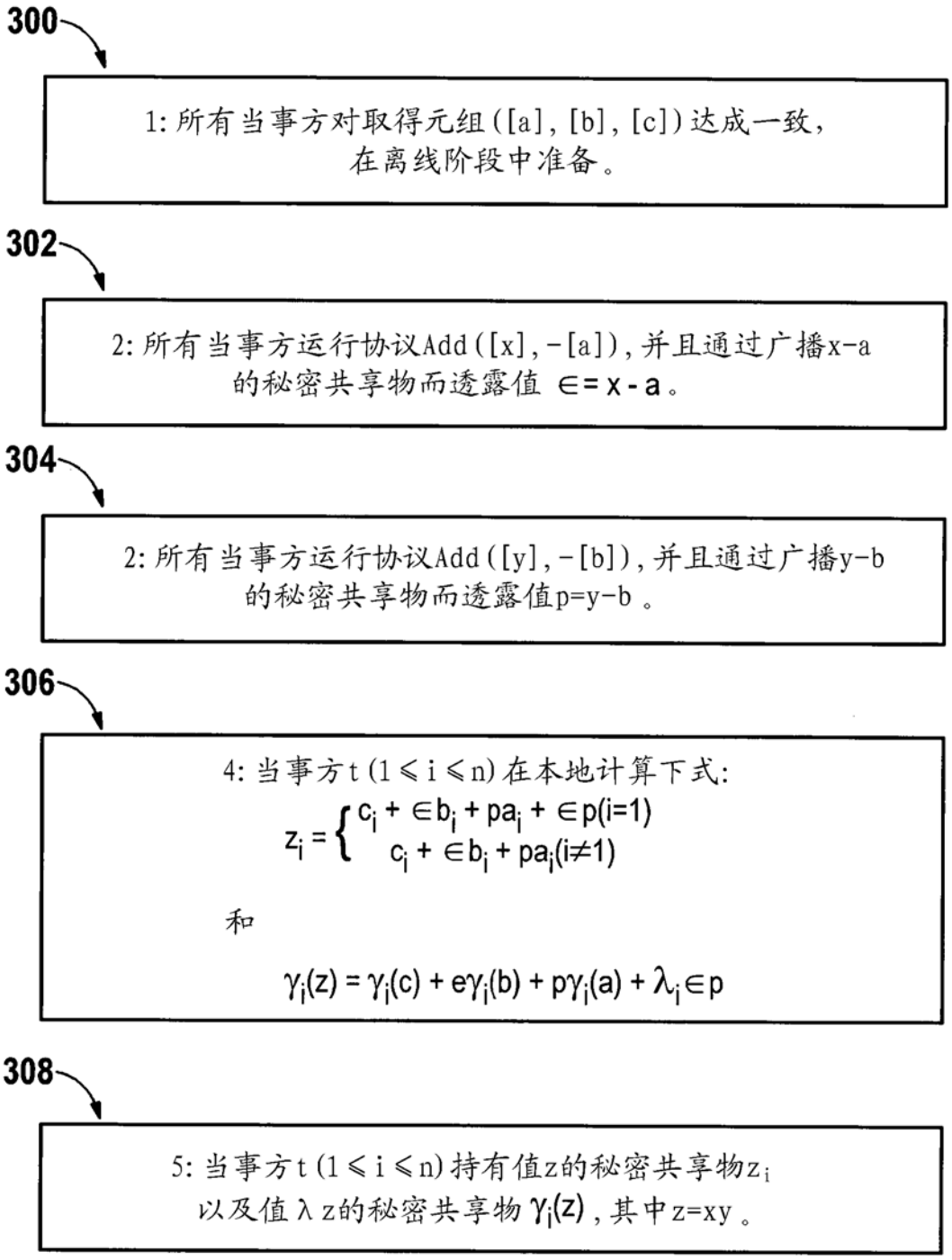

300
1: 所有当事方对取得元组([a], [b], [c])达成一致，
在离线阶段中准备。
302
2: 所有当事方运行协议Add([x], -[a])，并且通过广播x-a
的秘密共享物而透露值 ∈= x - a。
304
2: 所有当事方运行协议Add([y], -[b])，并且通过广播y-b
的秘密共享物而透露值p=y-b 。
306
4: 当事方t (1 ⩽ i ⩽ n)在本地计算下式:
$z_i = \begin{cases} c_i + \in b_i + pa_i + \in p(i=1) \\ c_i + \in b_i + pa_i (i \neq 1) \end{cases}$
和
$\gamma_i(z) = \gamma_i(c) + e\gamma_i(b) + p\gamma_i(a) + \lambda_i \in p$
308
5: 当事方t (1 ⩽ i ⩽ n)持有值z的秘密共享物z_i
以及值λz的秘密共享物 $\gamma_i(z)$，其中z=xy 。

图 3

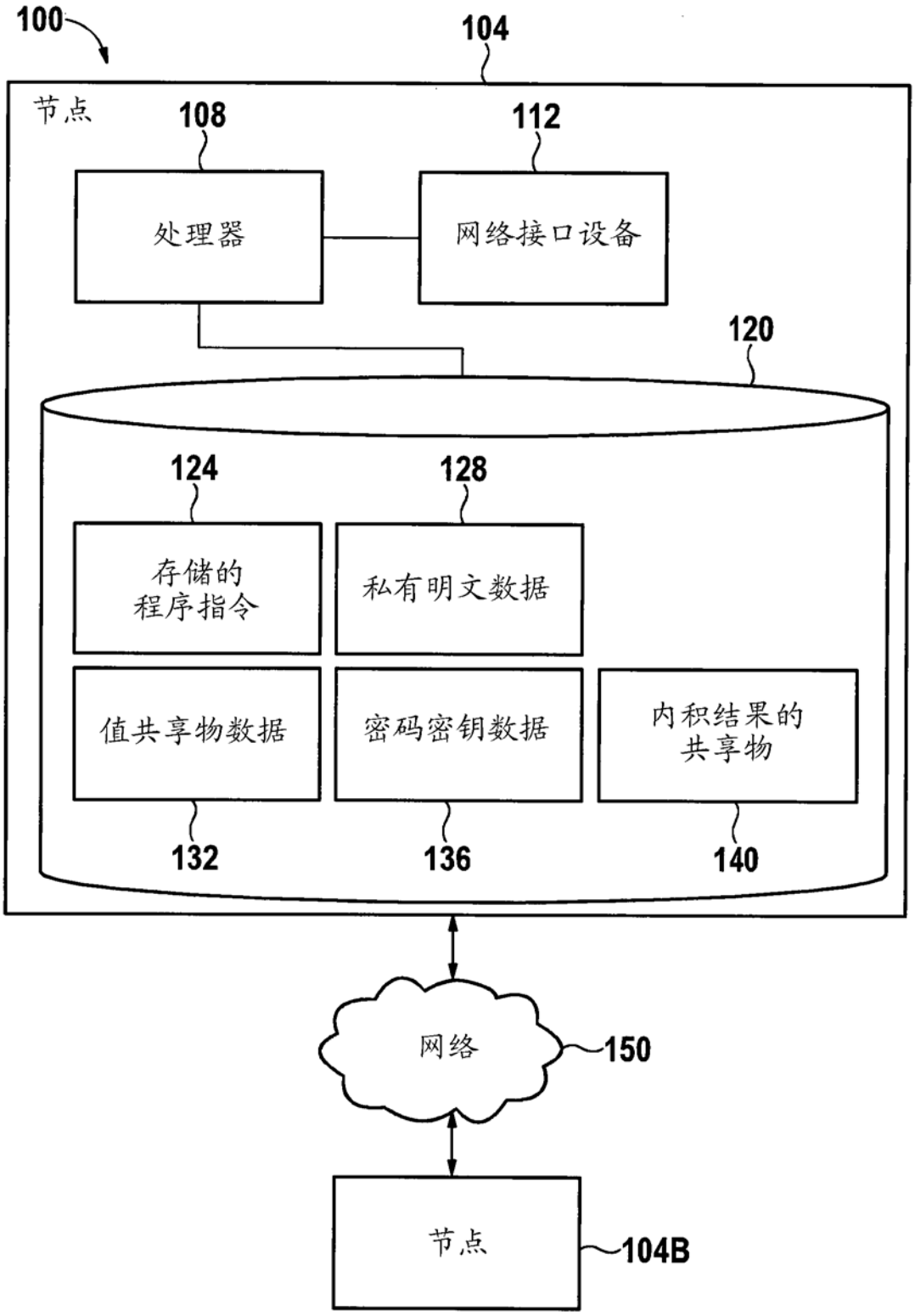

100
104
节点
108
112
处理器
网络接口设备
120
124
128
存储的
程序指令
私有明文数据
值共享物数据
密码密钥数据
内积结果的
共享物
132
136
140
网络
150
节点
104B

图 4

for 持有秘密共享物 a_1 的每个当事方 P_1, $1 \le i \le n$, do
Let $\Psi = \{P_1,\ldots,P_{t-1},P_{t+1},\ldots,P_n\}$ 是由超越本身的所有当事方组成的集合，并且Ω是具有
t个当事方的Ψ的所有子集的集合。
for 子集 $S = \{P_{k1},\ldots,P_{kt}\} \in \Omega$, 当事方 P_1 do
让 “Label” 是唯一标识S（例如，按顺序连结t个当事方）的字符串
随机均匀选择 $a_{1,k1,Label},\ldots,a_{t,kt-1,Label}$ ，并且让 $a_{t,kt,Label}$ 是使得
$a_{1,k1,Label}+\ldots+a_{t,kt,Label}=a_1$ 的值
经由认证的机密信道将 $a_{1,k1,Label}$, $(1 \le j \le t)$ 和 “Label” 发送到当事方 P_{kj}, ($k_j \in S$)
end
end
for 每个当事方 P_1, $1 \le i \le n$, do
P_1 将关于 “Label” 的重新分配的秘密共享物加起来，其由 $AggSet_1 = \{Label, \Sigma_{Label}\}$ 标示，
其中 “Label” 是唯一标识子集的字符串，并且 Σ_{Label} 是关于标注 “Label” 的重新分配的
秘密共享物的总和。应指出，P_1 还持有 a_1 的秘密共享物。
end

图 5

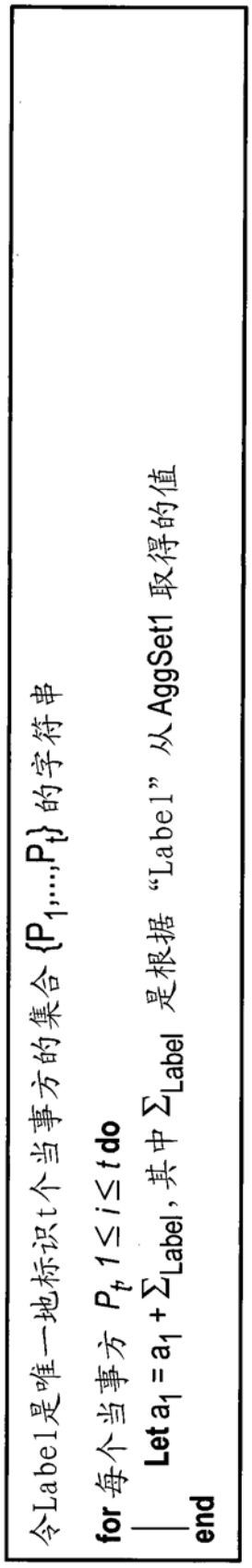

令Label是唯一地标识t个当事方的集合 {P1,...,Pt} 的字符串
for 每个当事方 Pi 1 ≤ i ≤ t do
Let a1 = a1 + ΣLabel，其中 ΣLabel 是根据 “Label” 从AggSet1 取得的值
end

图 6

当事方P_1	当事方P_2	当事方P_3	当事方P_4
x_1	x_2	x_3	x_4
	$(Label_{2,3} = "2\|\|3", x_{1,2,Label2,3})$	$(Label_{2,3} = "2\|\|3", x_{1,3,Label3,4})$	
	$(Label_{2,4} = "2\|\|4", x_{1,2,Label2,4})$		$(Label_{2,4} = "2\|\|4", x_{1,4,Label2,4})$
		$(Label_{3,4} = "3\|\|4", x_{1,3,Label3,4})$	$(Label_{3,4} = "3\|\|4", x_{1,4,Label3,4})$

图 7

当事方 P_1	当事方 P_2	当事方 P_3	当事方 P_4
x_1	x_2	x_3	x_4
	$(Label_{2,3} = "2\|\|3", x_{1,2,Label2,3})$ $(Label_{2,4} = "2\|\|4", x_{1,2,Label2,4})$	$(Label_{2,3} = "2\|\|3", x_{1,3,Label3,4})$ $(Label_{3,4} = "3\|\|4", x_{1,3,Label3,4})$	$(Label_{2,4} = "2\|\|4", x_{1,4,Label2,4})$ $(Label_{3,4} = "3\|\|4", x_{1,4,Label3,4})$
$(Label_{1,3} = "1\|\|3", x_{2,1,Label1,3})$ $(Label_{1,4} = "1\|\|4", x_{2,3,Label1,4})$		$(Label_{1,3} = "1\|\|3", x_{2,3,Label1,3})$ $(Label_{3,4} = "3\|\|4", x_{2,3,Label3,4})$	$(Label_{1,4} = "1\|\|4", x_{2,4,Label1,4})$ $(Label_{3,4} = "3\|\|4", x_{2,4,Label3,4})$
$(Label_{1,2} = "1\|\|2", x_{3,2,Label1,2})$ $(Label_{1,4} = "1\|\|4", x_{3,1,Label1,4})$	$(Label_{1,2} = "1\|\|2", x_{3,1,Label1,2})$ $(Label_{2,4} = "2\|\|4", x_{3,2,Label2,4})$		$(Label_{1,4} = "1\|\|4", x_{3,4,Label1,4})$ $(Label_{2,4} = "2\|\|4", x_{3,4,Label2,4})$
$(Label_{1,2} = "1\|\|2", x_{4,2,Label1,2})$ $(Label_{1,3} = "1\|\|3", x_{4,2,Label1,3})$	$(Label_{1,2} = "1\|\|2", x_{4,1,Label1,2})$ $(Label_{2,3} = "2\|\|3", x_{4,3,Label2,3})$	$(Label_{1,3} = "1\|\|3", x_{4,3,Label1,3})$ $(Label_{2,3} = "2\|\|3", x_{4,3,Label2,3})$	

图 8

当事方	持有秘密共享物	重新分配的秘密共享物 AggSet
当事方 P_1	x_1	$(Label_{1,2} = "1\|\|2", x_{3,1,Label1,2} + x_{4,1,Label1,2})$ $(Label_{1,3} = "1\|\|3", x_{2,1,Label1,3} + x_{4,1,Label1,3})$ $(Label_{1,4} = "1\|\|4", x_{2,1,Label1,4} + x_{3,1,Label1,4})$
当事方 P_2	x_2	$(Label_{1,2} = "1\|\|2", x_{3,2,Label1,2} + x_{4,2,Label1,2})$ $(Label_{2,3} = "2\|\|3", x_{1,2,Label2,3} + x_{4,2,Label2,3})$ $(Label_{2,4} = "2\|\|4", x_{1,2,Label2,4} + x_{3,2,Label2,4})$
当事方 P_3	x_3	$(Label_{1,3} = "1\|\|3", x_{2,3,Label1,3} + x_{4,3,Label1,3})$ $(Label_{2,3} = "2\|\|3", x_{1,3,Label2,3} + x_{4,3,Label2,3})$ $(Label_{3,4} = "3\|\|4", x_{2,3,Label3,4} + x_{4,3,Label3,4})$
当事方 P_4	x_4	$(Label_{1,4} = "1\|\|4", x_{2,4,Label1,4} + x_{3,4,Label1,4})$ $(Label_{2,4} = "2\|\|4", x_{1,4,Label2,4} + x_{3,4,Label2,4})$ $(Label_{3,4} = "3\|\|4", x_{1,4,Label3,4} + x_{2,4,Label3,4})$

图 9

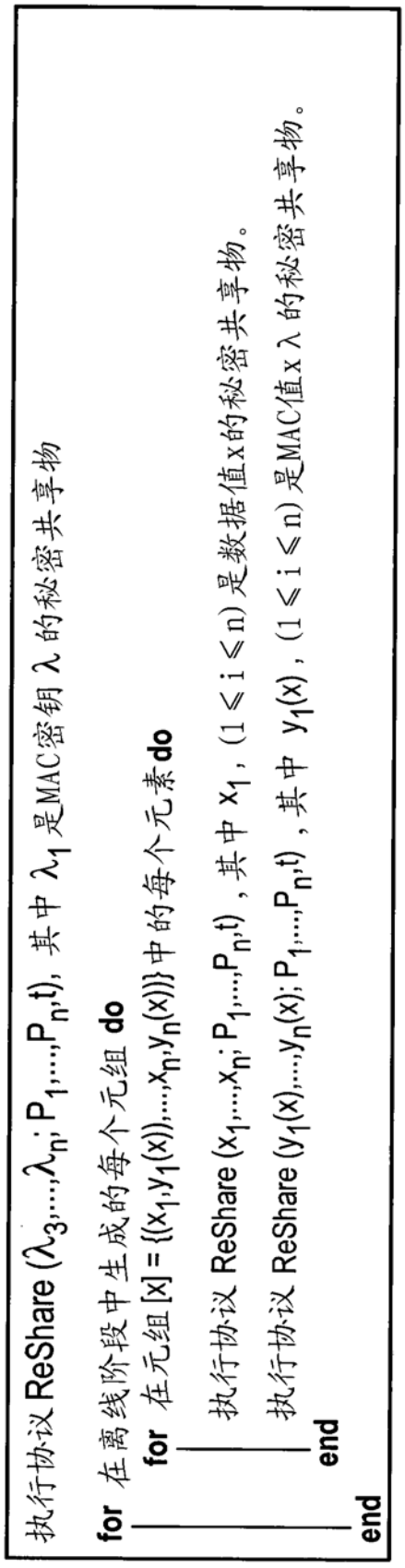

执行协议 ReShare (λ_3,...,λ_n; P_1,...,P_n,t), 其中 λ_1 是MAC密钥 λ 的秘密共享物
for 在离线阶段中生成的每个元组 do
for 在元组 [x] = {(x_1,$y_1(x)$),...,(x_n,$y_n(x)$)} 中的每个元素 do
执行协议 ReShare (x_1,...,x_n; P_1,...,P_n,t)，其中 x_1，($1 \leqslant i \leqslant n$) 是数据值x的秘密共享物。
执行协议 ReShare ($y_1(x)$,...,$y_n(x)$; P_1,...,P_n,t)，其中 $y_1(x)$，($1 \leqslant i \leqslant n$) 是MAC值x λ 的秘密共享物。
end
end

图 10

执行协议 ReCombine (λ_3,$AggSet_1$,...,λ_t,$AggSet_t$; P_1,...,P_t), 使得当事方P_t获得MAC密钥 λ的秘密共享物

for 在离线阶段中生成的每个元组 **do**

  for 持有 (x_1,$AggSet_1$,$y_1(x)$,$AggSet_n$)的每个当事方P和元组中的每个元素 **do**

    执行协议 ReCombine (x_1,$AggSet_1$,...,x_t,$AggSet_t$; P_1,...,P_t), 使得当事方P_t获得数据值x的秘密共享物。

    其中 x_1, ($1 \leqslant i \leqslant n$)是数据值x的秘密共享物。

    执行协议 ReCombine ($y_1(x)$,$AggSet_1$,...,$y_t(x)$,$AggSet_t$; P_1,...,P_n,t), 使得当事方 P_1 获得MAC值x λ 的秘密共享物。

  end

end

图 11

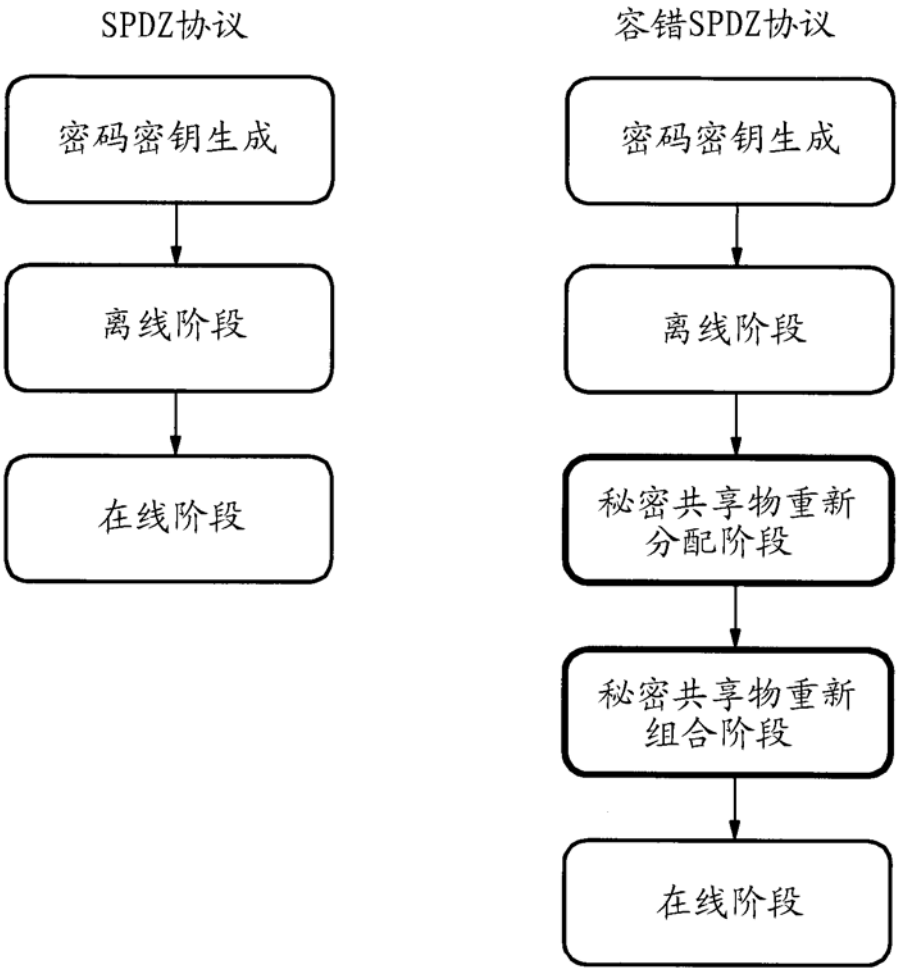


图 12

	SPDZ	容错SPDZ
存储消耗	1	$\binom{n-1}{t}+1$
恶意当事方数量	n-1	t-1
容错	0	n-t

图 13