TREE-BASED CIPHERTEXTS COMPARISON APPROACH

基于树的密文比较方法

1. Scenarios(情景)

假设有一个用户（U）和一个服务器（R）。U有许多整数，这些整数先用一种加密算法加密，然后传给R。 U向R请求来获得一个和不超过θ的整数对（x，y）。注意，除了大于、相等或小于关系之外，x、y和θ的明文不能向R公开。一种简单的方法是下载所有整数，在本地计算求和，并选择满足约束的整数对。然而，如果想要将这种计算方法加载到云端，这个方法就没有意义了。因此，应该找一个实际的办法来解决这一问题。

请注意，这种情形和众所周知的SMC方案是不同的。在SMC的设置当中，一组（两个或两个以上）具有私人投入的缔约方希望根据其投入来进行函数计算，同时除了函数的结果外什么都不显示。该函数用于许多实际应用，如外汇市场。SMC是一个协作计算问题，解决了一组互相不信任的参与者之间的隐私保护问题。

所有整数对（x,y）和约束条件（θ）的密文都上传到云服务器，云服务器来负责不等式测试。此外，我们还可以将两个密文的和与另一个密文之间的关系告诉服务器，这在文献[17]中被称为受控披露。

似乎我们可以利用同态加密技术，因为它支持计算x+y时的加操作，不过由于同态加密是概率加密，因此我们无法通过x+y和θ的密文来确定他们之间的关系。

1. Main Idea(主要思想)

基于树的密文比较协议的主要思想是用ORE原语对整数进行编码。根据我们目前的知识来看，没有方法能同时支持ORE和同态性质。因此，我们设计了一种新的方法来解决这个问题，这是由下列事实推动的：

如果我们想比较x+y和θ，我们可以比较x和θ/2以及y和θ/2。通过这两种对应关系的组合我们可以得到四种结果：

如果x>θ/2（x≤θ/2）和y>θ/2（y≤θ/2），我们可以知道x+y>θ（x+y≤θ）。在剩下的两个案例中，即x>θ/2和y<θ/2，或x≤θ/2和y≥θ/2，我们无法获获得确定性结果。这时，我们可以进一步将θ/2分为θ/4。然后我们可以比较x，y与θ/4和3θ/4。

通过迭代执行这样的操作，我们可以越来越确定确定x+y和θ之间的关系。由于ORE性质，上述密文操作很容易便可以执行。接下来，我们将展示如何通过使用树结构有效地实现这个想法。C. Details of Protocol（协议细节）

为了实现通过密文来比较x+y和θ，我们构造了一个成本约束树，它的节点表示与θ相关的特定值。为了清楚起见，我们定义E（m）为m的ORE密文。

树结构的示例如图6所示。

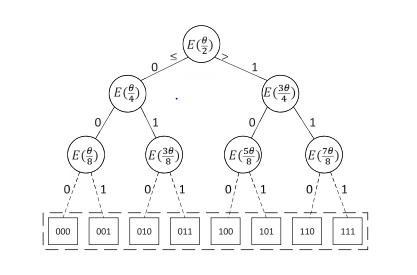
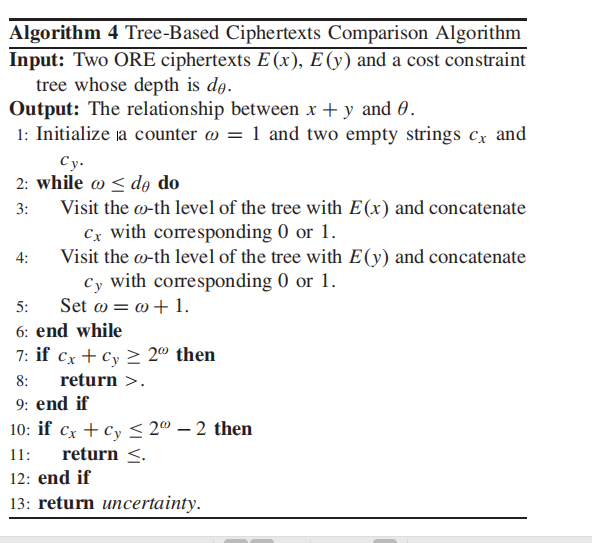


Fig. 6.

对于每个节点，我们将它的左子树定义为0，右子树定义为1。如果一个整数不大于该节点的值，则与左子路径进行进一步比较；否则，与右子树。因此，自根节点到叶节点的任一路径，我们可以得到一个路径代码，有效的显示了比较过程。例如，传入一个大小为5θ/16的整数，它将经过节点E（θ/2）、E（θ/4）和E（3θ/8），从而以代码010结束。我们定义了路径代码的长度（位数）为β。β实际上等于以dθ表示的树的深度。

现在我们可以通过如下过程来确定x+y和θ之间的关系。我们首先得到x和y的ORE密文，以及它们的路径代码cx和cy（通过遍历树得到）。当计算cx+cy时，如果发生溢出（即cx+cy≥2β），我们知道x+y>θ。如果cx+cy≤2β−2，我们也知道x+y≤θ否则，我们无法确定关系，最终以不确定性结束。

我们总结一下算法4中的这个过程

****